HIGHWAY INFORMATION SYSTEM
VOLUME 1: USER INFORMATION

Department of Civil Engineering & Engineering Mechanics Montana State University Bozeman

HIGHWAY INFORMATION SYSTEM

VOLUME 1: USER INFORMATION

Prepared for the

STATE OF MONTANA
DEPARTMENT OF HIGHWAYS
PLANNING AND RESEARCH BUREAU

In cooperation with the

U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL HIGHWAY ADMINISTRATION

The opinions, findings and conclusions expressed in this publication are those of the authors and not necessarily those of the Montana Department of Highways or the Federal Highway Administration

Prepared by

Glen L. Martin, Larry J. Coats, and Michael J. Meldahl

DEPARTMENT OF CIVIL ENGINEERING AND ENGINEERING MECHANICS
MONTANA STATE UNIVERSITY
Bozeman, Montana 59715

June, 1972

FOREWORD

This report is a documentation of the highway data bank research undertaken by the Department of Civil Engineering and Engineering Mechanics, Montana State University. The research was sponsored by the Montana Department of Highways in cooperation with the U.S. Department of Transportation, Federal Highway Administration.

Conceptually, the CE & EM Department was responsible for developing an information retrieval system for rapid access to highway data. Specifically, the responsibility was to produce the Roadlog, Traffic by Sections, Accident by Sections and Sufficiency by Sections reports as a direct application of the system. In addition, preliminary investigation of the feasiblity of a geometrics file and a preliminary investigation of the storage and retrieval of visual images was included in the project objectives.

In light of the foregoing, it is desirable to present the report in two volumes: Highway Information System Volume 1: User Information, and Highway Information System Volume 2: Programmer Information. Volume 1 deals with the use of the system, including information on data coding and on the execution of programs within the system. Volume 2 deals with the detailed operation of the system, providing information on the modification of programs existing within the system as well as on the addition of programs to the system. Volume 1 is a prerequisite publication to Volume 2.

In developing the system, the CE & EM Department has had the privilege of using an IBM OS 360/40 computer located at the Data Processing Bureau of the Montana Department of Highways in Helena. PL/I has been used as the programming language for nearly all of the HIS routines because of its versatility in input-output (I/O) and interchangeability of files. BAL (assembler) has been used for several routines because of its increased capabilities and efficiency over other languages.

The project could never have progressed to its current state were it not for the continual encouragement from and the patient, sustained assistance of both the Planning and Research Bureau and the Data Processing Bureau of the Montana Department of Highways, and of the Montana State Highway Patrol.

The project conclusion was also hastened by the significant effort of other project personnel: Francis C. F. Yu, Leroy R. Zook, Philip A. House,

Alfred C. Scheer, Paul W. Burkhart, Robert C. Smith, Harry E. Hughes, Ronald E. Billstein, Daniel D. Urbach and Donald R. Reichmuth. Their assistance has been invaluable.

HIGHWAY INFORMATION SYSTEM
VOLUME 1: USER INFORMATION

TABLE OF CONTENTS

Chapter		Page
1-I	INTRODUCTION	1
	The Supervisor	3
	The Reference Post System	3
	The Roadlog Subsystem	4
	The Traffic and True Mileage Subsystem	6
	The Accident Subsystem	10
	The Sufficiency Subsystem	16
	Summary	16
1-II	ROADLOG USER INFORMATION	18
	Introduction	18
	Roadlog Mileage Record Coding	20
	Roadlog Descriptor Record Coding	20
	OS JCL Statements	36
	HIS Commands	36
	Continuation cards	36
	Comment cards	37
	Examples of valid HIS commands	37
	Examples of invalid HIS commands	37
	The DATA parameter	37
	Report and report summary commands	38
	LIST-&-SUM COMMAND	39
	SURF-TYPE COMMAND	40
	SUMMARY-BY-ROUTES COMMAND	41
	SUMMARY-BY-LOCATION COMMAND	41
	FORHWY-SUMMARY COMMAND	42
	SUM-LOOPS-&-SPURS COMMAND	42
	Formatting option parameters	43
	FORMAT PARAMETER	43
	PAGE-NUMBER PARAMETER	44
	TABLE-NUMBER PARAMETER	44
	TOP-MARGIN PARAMETER	45
	PAGE-EJECT PARAMETER	45
	Roadlog file maintenance	46
	DUMP COMMAND	46
	LIST COMMAND	47
	LIST-ILOOPS COMMAND	47
	UPDATE COMMANDS	48
	COPY COMMAND	49
	CREATE COMMAND	50
	Storage of the Roadlog data	51

Chapter		Page
1-III	TRAFFIC AND TRUE MILEAGE USER INFORMATION	52
	The Traffic File	52
	Traffic Count Records	52
	Traffic Descriptor Records	56
	Coding Traffic Sections	56
	Defining traffic sections	56
	Rural and out-of-state sections	59
	Municipal sections	60
	Non-existent sections	60
	The True Mileage File	61
	OS JCL Statements	61
	HIS Commands	61
	Examples of valid HIS commands	61
	Report and report summary commands	63
	CREATE-TRAFSUB COMMAND	63
	LIST-TRAFSUB COMMAND	64
	TRAFFIC-BY-SECTIONS COMMAND	64
	SUMMARY-BY-ROUTES COMMAND	65
	Formatting option parameters	65
	FORMAT=NOREDUCE PARAMETER	65
	PAGE-NUMBER PARAMETER	66
	Traffic file maintenance commands	66
	DUMP COMMAND	66
	LIST COMMAND	66
	UPDATE COMMAND	67
	COPY COMMAND	69
	CREATE COMMAND	69
	UPDATE-BY-YEAR COMMAND	70
	True Mileage file maintenance commands	70
	LIST COMMAND	70
	UPDATE COMMAND	71
	COPY COMMAND	72
	CREATE COMMAND	73
	Storage of the Traffic and True Mileage data	73
1-IV	ACCIDENT USER INFORMATION	74
	The Accident Files	74
	Accident Detail Record Coding	74
	Accident Vehicle Record Coding	74
	OS JCL Statements	99
	HIS Commands	99
	Accident file maintenance procedures	99
	EDIT-DATA-CARDS COMMAND	99
	UPDATE-ERRORS COMMAND	104
	STORE-DATA-CARDS COMMAND	106
	LOAD-ACCIDENT-DATA COMMAND	107
	MERGE-ACCIDENT-FILES COMMAND	107

Table of Contents (continued)

Chapter		Page
1-IV (Cont'd)	Summary commands	108
	PRINT-MEMOS COMMAND	108
	NATIONAL SAFETY COUNCIL FORM 16 COMMAND	109
	Report commands	109
	CREATE-DIRECTORY and LOAD-ACC-DIRECTORY COMMANDS	110
	CREATE-ACCSUB COMMAND	111
	CREATE-ACC-LIMITS COMMAND	112
	ACCIDENT-BY-SECTIONS and MULTIPLE-ACC-LOCNS COMMANDS	112
	Storage of the Accident Data	113
1-V	SUFFICIENCY USER INFORMATION	114
	Introduction	114
	Sufficiency Rating Record Coding	114
	Sufficiency Descriptor Record Coding	114
	OS JCL Statements	118
	HIS Commands	119
	Report and summary commands	119
	CREATE-SUFFSUB,PHASE=SUFFICIENCY COMMAND	119
	CREATE-SUFFSUB,PHASE=ROADLOG COMMAND	120
	CREATE-SUFFSUB,PHASE=TRAFFIC COMMAND	121
	CREATE-SUFFSUB,PHASE=ACCIDENT COMMAND	123
	CREATE-SUFFSUB,PHASE=CALCULATION COMMAND	124
	LIST-BY-SECTION COMMAND	126
	LIST-BY-DISTRICT COMMAND	126
	LIST-BY-RATING COMMAND	127
	MAP-TABLES COMMAND	128
	RATING-BY-DISTRICT COMMAND	128
	DEF-MILES-BY-COUNTY COMMAND	129
	Sufficiency file maintenance commands	129
	UPDATE COMMAND	129
	COPY COMMAND	131
	CREATE COMMAND	132
1-VI	PRELIMINARY STUDY OF RETRIEVAL OF ROADWAY GEOMETRIC INFORMATION	133
	Introduction	133
	Roadway Locations	133
	The reference post system	133
	The stationing system	134
	Relationship between the reference post and stationing systems	135
	Geometrics Files	135
	The horizontal file	135
	The vertical file	137

Table of Contents (continued)

Chapter		Page
1-VI (Cont'd)	Geometrics Programs	139
	Vertical program	139
	Horizontal program	140
	Special conditions for the programs	140
	Geometrics Test Section	140
	Problems encountered	142
	Recommendations Concerning Geometrics File	143
1-VII	STORAGE AND RETRIEVAL OF VISUAL INFORMATION	145
	Introduction	145
	Microfilm Methods	146
	Roll/sheet/aperture card microfilm	147
	"Keyed microfilm"	148
	Central microfilm with remote access	148
	Computer Graphics Methods	149
	Summary	150

LIST OF FIGURES

Figure No.		Page
1-I-1	HIS organization	2
1-II-1	Hypothetical stretch of highway	19
1-V-1	Traffic counts within a sufficiency section	122
1-VI-1	Sketch of test section	141

LIST OF TABLES

Table No.		Page
1-I-I	DATA ELEMENTS OF ROADLOG FILE RECORDS	5
1-I-II	DATA ELEMENTS OF TRAFFIC FILE RECORDS	7
1-I-III	DATA ELEMENTS OF TRUE MILEAGE FILE RECORDS	8
1-I-IV	DATA ELEMENTS OF TRAFFIC SUMMARY FILE RECORDS	9
1-I-V	DATA ELEMENTS OF ACCIDENT DETAIL FILE RECORDS	11
1-I-VI	DATA ELEMENTS OF ACCIDENT VEHICLE FILE RECORDS	12
1-I-VII	DATA ELEMENTS OF ACCIDENT DIRECTORY FILE RECORDS	13
1-I-VIII	DATA ELEMENTS OF ACCIDENT REPORT FILE RECORDS	14
1-I-IX	DATA ELEMENTS OF ACCIDENT LIMITS FILE RECORDS	15
1-I-X	DATA ELEMENTS OF SUFFICIENCY FILE RECORDS	17
1-II-I	DATA ELEMENTS AND CODING INFORMATION FOR ROADLOG MILEAGE RECORDS	21
1-II-II	ROUTE SYSTEM CODES	23
1-II-III	PROJECT NUMBER CODES	24
1-II-IV	REMARK CODES	25
1-II-V	COUNTY NAME CODES	26
1-II-VI	MILEAGE LOCATION CODES	27
1-II-VII	POPULATION CODES	28
1-II-VIII	CITY NAME CODES	29
1-II-IX	SURFACE TYPE CODES	30
1-II-X	DATA ELEMENTS AND CODING INFORMATION FOR ROADLOG DESCRIPTOR RECORDS	31
1-II-XI	EXPLANATION OF DESCRIPTION FIELD OF ROADLOG DESCRIPTOR CARDS	32
1-II-XII	FORMAT OF DESCRIPTION FIELD FOR "CO" DESCRIPTOR RECORDS	33
1-II-XIII	FORMAT OF DESCRIPTION FIELD FOR "IL" DESCRIPTOR RECORDS	35
1-III-I	DATA ELEMENTS AND CODING INFORMATION FOR TRAFFIC COUNT RECORDS	53
1-III-II	REMARK CODES	55
1-III-III	DATA ELEMENTS AND CODING INFORMATION FOR TRAFFIC DESCRIPTOR RECORDS	57
1-III-IV	DATA ELEMENTS AND CODING INFORMATION FOR TRUE MILEAGE RECORDS	62

CHAPTER 1-I

INTRODUCTION

HIS (Highway Information System) consists of a set of programs and a number of data files. The programs allow file maintenance, and the production of reports and summaries from the files.

A design criterion of HIS has been to locate each data item in one and only one location. This promotes data integrity, forces compatibility among the files, and simplifies updating.

Because each data item is in only one location, it is often necessary to access two or more files in the production of summaries. In order to allow this cross-referencing of files, a common referencing method is required. All of the files in HIS are "keyed," based on the reference post system of locating points on roadways. By organizing all of the files around this key, the files may be successfully cross-referenced to bring together data from several files.

A further design criterion of HIS has been to orient the system toward the user. To this end, a "supervisor" has been implemented. The supervisor reads "commands" coded by the user, and executes the commands. Through the use of this supervisor, the system may be used by persons unfamiliar with the internal workings of HIS.

HIS is designed to run in conjunction with the IBM System/360 Operating System (OS). Figure 1-I-1 shows the relationship between the HIS components and OS.

As can be seen from Figure 1-I-1 user input falls into three categories: 1) OS Job Control Language (JCL), 2) commands to the HIS supervisor, and 3) input data to HIS routines.

OS JCL is required in order to instruct the Operating System to execute the HIS supervisor. HIS commands are required to define the operations to be performed, and to execute the proper HIS routines. Input data is required by some of the HIS routines in order to update HIS data files.

The following signs and conventions, consistent with those used by IBM, have been adopted throughout these manuals:

1) Uppercase letters and punctuation marks (except for brackets and braces) must be coded.

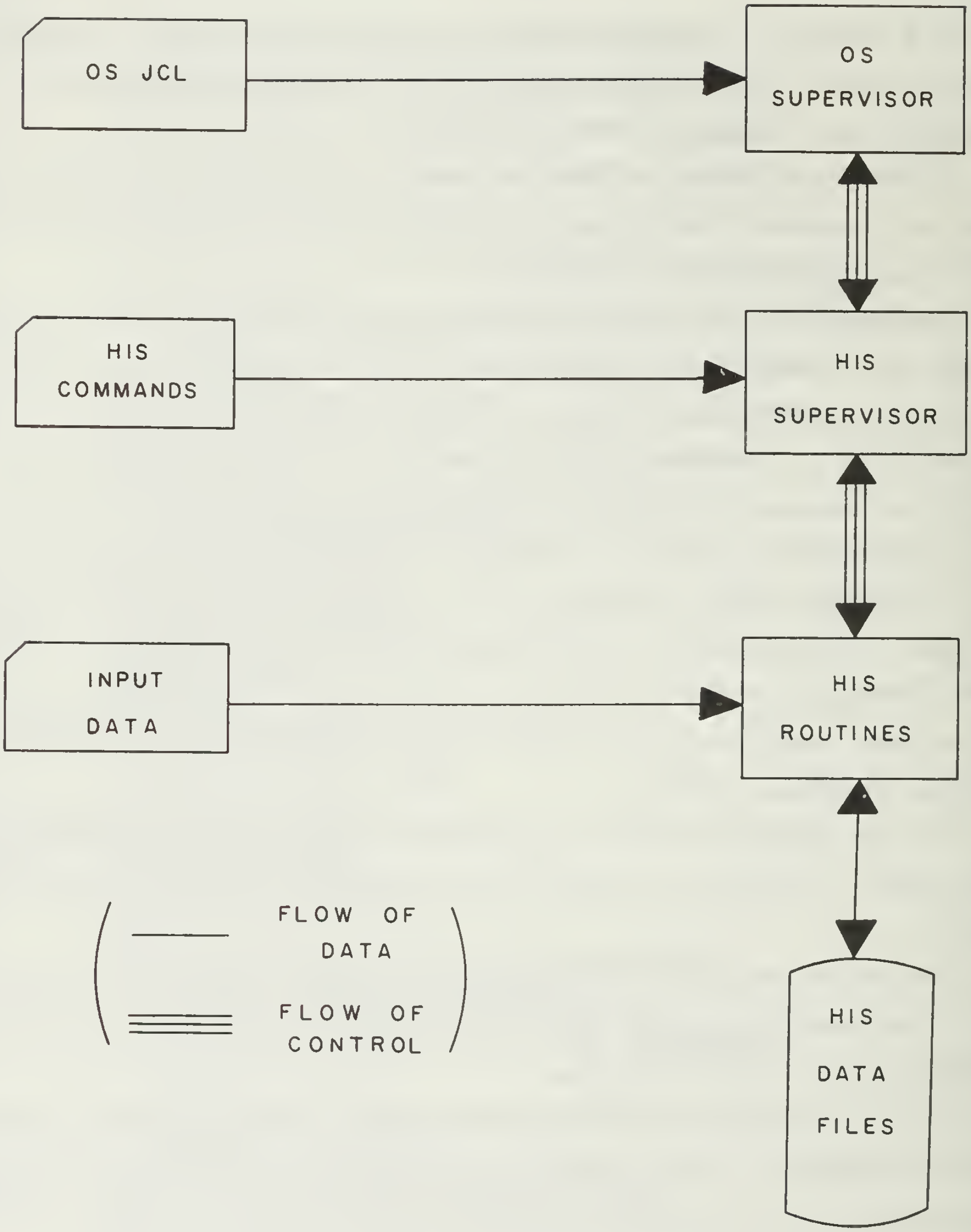


Figure 1-I-1. HIS organization.

2) Lowercase letters and terms represent information that must be supplied by the user.

3) Information contained within brackets [ ] represents an option that may be included or omitted, depending upon the requirements of the user.

4) Options contained within braces { } represent alternatives, one (and only one) of which must be chosen.

The Supervisor

In order to provide all of the file maintenance and report generation capabilities required for a large data bank system, a sizeable number of separate programs must be available for execution. A "supervisor" routine, able to process simplified commands coded by the user, has been included in the system to aid in selecting the proper program for a particular application. The supervisor can be likened to an automatic record selector (jukebox): the user indicates his need by selecting a song title and through machine control mechanisms the jukebox selects the proper record and plays it. Similarly, the user of the HIS supervisor indicates his needs by means of a command card, and the supervisor selects one of the many available routines to carry out the command.

The Reference Post System

The reference post system is a method for identifying roadway locations (milepoints). The system consists of a set of non-uniformly spaced physical reference posts located along the roadways. The reference posts, in general, are approximately a mile apart, but may vary considerably from this distance. The first marker of a route is numbered zero, the succeeding markers are numbered sequentially.

In order to uniquely specify a roadway location on a route (a milepoint), two items are specified: the number of a reference post, and the distance from that reference post to the roadway location. The distance may be positive or negative; it is positive if travel from the reference post to the roadway location is toward higher-numbered reference posts, and negative if travel is toward lower-numbered reference posts.

As an example of the use of the reference post system, a point located 0.348 miles beyond reference post number 146 toward reference post 147 is specified as milepoint 146+0.348. The point may also be referenced in relation to marker number 147. If, for example, markers 146 and 147 are 1.459 miles apart, the point may be specified as 147-1.111.

In order to process two or more locations, which are specified by different reference posts, it is necessary to locate them by means of a common point. In order to do this, a "true mileage" file is built which locates each reference post with respect to the beginning point of the route. Through the use of this file, it is possible to locate every point on the roadway with respect to the beginning of the route. Hence, it is also possible to determine the distance between any two points on the roadway.

To complete the key for HIS files, the route system and Federal Aid route number are concatenated (joined together in sequence) with the location on the route (reference post and distance). The route system is a 1-character code, "I" for Interstate, "P" for Primary, and "S" for Secondary. The Federal Aid route number is a 3-digit number.

The complete key provides a unique identification for every roadway location on the Federal Aid system. This identification method is used throughout the HIS files in order that the files may be kept compatible.

The Roadlog Subsystem

The Roadlog subsystem is based upon a single raw data file: the Roadlog file. This file contains physical roadway information pertaining to all Federal Aid Interstate, Primary, and Secondary routes. The data elements of the file are shown in Table 1-I-I.

In addition to the programs required for Roadlog file maintenance, programs within the Roadlog subsystem print a number of summaries for inclusion in the annual Federal Aid Roadlog report.

The Roadlog file is organized around the common "key" of HIS -- the Federal Aid route system, Federal Aid route number, reference post, and distance from the reference post. Each record contains the key of a Roadlog section break -- a location at which a discontinuity occurs, such as a change in the physical characteristics of the roadway, a county line, a city limit,

TABLE 1-I-I

DATA ELEMENTS OF ROADLOG FILE RECORDS

Items

Key (Federal Aid Route System, Federal Aid Route Number, Reference Post, and Milepoint).

Remark.

Section Length.

Route Length.

Constructed Length.

Unimproved Length.

Wye Length.

Description.

Project Number.

Divided/Undivided Code.

Number of Lanes.

Population Code.

City Number.

County Number.

Year Built.

Year Improved.

Forest Highway Number.

Administration Code.

Location Codes.

Project Class.

Surface Width.

Roadway Width.

Surface Thickness.

Base Thickness.

Surface Type Code.

Surface Type.

Maintenance Section.

Date.

a junction, etc., -- and defines the section of road between that point and the next.

When two routes are coincident, a cross-reference technique is used to prevent duplication of data in the file. All of the sections within the coincident stretch are fully coded for one of the routes. The second route contains a "coincident" record, specifying the beginning and ending mile-points on the first route of the coincident roadway. Any programs processing the second route can, when the coincident record is read, access the applicable records of the first route to obtain the data for the coincident section.

The Roadlog subsystem is not dependent on any of the other HIS subsystems, as all Roadlog summaries are taken from only the Roadlog file. However, all of the other subsystems rely on the Roadlog subsystem for information. A workable communications system between the personnel working on the various systems is imperative, because all of the files must be kept fully compatible in order to obtain accurate summaries and reports.

The Traffic and True Mileage Subsystem

The Traffic and True Mileage subsystem is based on three raw data files: the Roadlog, True Mileage, and Traffic files. The subsystem builds a "summary" file from the Traffic and True Mileage files for use in printing summaries. In particular, the summary file is used for producing the annual Traffic by Sections report. The data elements of the Traffic, True Mileage, and Traffic Summary files are shown in Tables 1-I-II, 1-I-III, and 1-I-IV, respectively. The Traffic file provides "average daily traffic" (ADT) counts at each count station on the Federal Aid system for three separate years. Each count station is located by its "key" (milepoint).

For the purpose of the construction of the Traffic Summary file (and hence the production of the Traffic by Sections report), count stations may be either "major" or "minor." The two types are distinguished by means of a "remark" code stored within the record. A stretch of highway between any two successive (major or minor) count stations is known as a "subsection." A stretch of road between two consecutive major count stations is a "section;" each section is composed of one or more subsections, depending upon the number of minor count stations within the section.

TABLE 1-I-II

DATA ELEMENTS OF TRAFFIC FILE RECORDS

Items

Key (Federal Aid Route System, Federal Aid Route Number, Reference Post, and Milepoint).

Actual/Estimated Code.

Remark Code.

Data for each of Three Years:

- Year.
- Average Daily Traffic.
- Percentage of Out of State Vehicles.
- Percentage of Pickups.
- Percentage of Commercial Vehicles.

Future Factor.

Design Hour Volume.

Date of Update.

TABLE 1-I-III

DATA ELEMENTS OF TRUE MILEAGE FILE RECORDS

Items
Federal Aid Route System.
Federal Aid Route Number.
Reference Post.
True Mileage.
Date of Update.

TABLE 1-I-IV

DATA ELEMENTS OF TRAFFIC SUMMARY FILE RECORDS

<u>Items</u>

Key.

Remark.

Section Length.

Data for each of Three Years:

Vehicle Miles--All Vehicles
Vehicle Miles--Out of State Vehicles
Vehicle Miles--Pickups
Vehicle Miles--Commercial Vehicles

In order to calculate the vehicle miles of a subsection, the ADT's at the count stations at each end of the subsection are averaged, and this average multiplied by the length of the subsection. The vehicle miles of a section is the summation of the vehicle miles of its subsections.

The "weighted ADT" for a section is the vehicle miles of the section divided by the length of the section.

The Traffic Summary file generated for the Traffic by Sections report contains the length of each station, and the vehicle miles of each section on all routes of the Federal Aid system. In addition, the total rural mileage and rural vehicle miles for each route and each route system is stored.

The Accident Subsystem

The accident subsystem consists of two files with accident data: a detail file with information pertaining to overall data on accidents (Table 1-I-V) and a vehicle file with data on persons and vehicles involved in the accidents (Table 1-I-VI). In addition, the subsystem utilizes the Roadlog, Traffic, and True Mileage files.

After the accident data are keypunched for entry into the files, a set of programs provides for editing of the data as it is read to help prevent faulty data from being added to the file. The new accidents, after editing, are formatted into appropriate form and merged with the existing files.

The subsystem has the capability of producing the annual Accident by Sections report and National Safety Council Form 16, on demand. Three summary files are built for the production of the Accident by Sections report. An Accident Directory file (Table 1-I-VII) is generated which allows access of accidents by milepoints. An Accident Report file (Table 1-I-VIII) is generated from the Roadlog, Traffic, True Mileage, and Accident Directory files. An Accident Limits file (Table 1-I-IX), containing upper and lower limits for accident rates, is then generated from the Accident Report file. From these three summary files, the Accident by Sections report is compiled and printed.

The Accident Directory file is also utilized by the Sufficiency subsystem.

TABLE 1-I-V

DATA ELEMENTS OF THE ACCIDENT DETAIL FILE RECORDS

Items

Key (Accident Number).

Date and Time Occurred.

Date and Time Notified.

Date and Time Arrived.

City Number (Municipal Accidents).

County Number.

Location.

First Harmful Event.

First Object Hit Off Roadway.

Injury Severity.

Damage Severity.

Class of Trafficway.

Roadway-Related Location.

Junction-Related Location.

Number of Vehicles.

Number of Pedestrians.

Number of Fatalities.

Number of Injuries.

Weather Condition.

Road Condition.

Light Condition.

Traffic Controls.

Other Damage--Type.

Other Damage--Severity.

Other Damage--Owner.

Posted Speed.

Engineering Study Requested.

Analysis (Contributing Circumstances).

Collision Type.

Reportable/Non-Reportable.

Investigated/Non-investigated.

TABLE 1-I-VI

DATA ELEMENTS OF ACCIDENT VEHICLE FILE RECORDS

Items

Key.

Name.

Driver's License.

State.

Birthday.

Re-examination Code.

Charge Code.

Summons Number.

Contributing Circumstances.

Alcohol--Driver and Passengers.

Sex--Driver and Passengers.

Injury Severity--Driver and Passengers.

Age--Driver and Passengers.

Vehicle Year.

Vehicle/Pedestrian Intent.

Vehicle Body Style.

Trailer Style.

Interstate Traffic.

Vehicle I. D. or License Number.

Damage of $250 or More.

Damage Severity.

TABLE 1-I-VII

DATA ELEMENTS OF ACCIDENT DIRECTORY FILE RECORDS

Items

Key.

Number of Fatalities.

Number of Injuries.

Date and Time.

First Harmful Event.

Collision Type.

Road Surface Condition.

Number of Lanes.

Date Flag.

TABLE 1-I-VIII

DATA ELEMENTS OF ACCIDENT REPORT FILE RECORDS

Items

Route System (I, P, or S).

Route Number.

Milepoint.

Remark.

Description.

Items for each of Three Years:

- Year.
- Average Daily Traffic.
- Number of Accidents.
- Number of Injury Accidents.
- Number of Fatality Accidents.
- Number of Injuries.
- Number of Fatalities.

Number of Lanes.

City Number.

TABLE 1-I-IX

DATA ELEMENTS OF ACCIDENT LIMITS FILE RECORDS

Items

Route System (I, P, or S).

Route Number.

Data for each of Three Years:

Upper Limit
Lower Limit

The Sufficiency Subsystem

The data elements of the Sufficiency file are shown in Table 1-I-X. The sufficiency subsystem also uses the files of the Roadlog, Traffic, and Accident subsystems in order to compile and print the annual Sufficiency by Sections report.

A summary file, the Sufficiency Report file, is generated from these files with the required data for each Sufficiency section. The summary file contains all of the information for each of the summaries in the Sufficiency by Sections report.

Summary

HIS consists of a set of programs, data files, and summary files. Present usage allows the production of a large number of summaries; many additional summaries are possible from the data existing in the files. All of the files are based around a common "key" (milepoint), allowing full access compatibility of the files.

The summary files contain intermediate results between the data files and reports which are printed by HIS. These files allow computations to be made once and the results saved for future reference.

All of the HIS files are fully compatible with teleprocessing applications available from IBM. Although it is not presently feasible (because of the existing computer system configuration) to utilize teleprocessing techniques in Montana, the teleprocessing capabilities may be added with little or no alteration to HIS.

TABLE 1-I-X

DATA ELEMENTS OF SUFFICIENCY FILE RECORDS

Items

Key.

Description.

Design Speed.

Terrain.

Average Speed.

Percent of Sight Distances less than Design.

Number of Stopping Sight Distances less than Design.

Number of Horizontal Curves Sharper than Design Degree of Curvature.

Number of Narrow Bridges.

Foundation Rating.

Surface Rating.

Drainage Rating.

Section Length--Under Construction and Non-Existent Sections.

Date of Update.

CHAPTER 1-II

ROADLOG USER INFORMATION

Introduction

The Roadlog file is a disk-resident file containing information about the physical characteristics of highways. Data is stored in the file, according to location on Federal Aid routes.

Any point on the center-line of a highway (a milepoint) may be located by specifying: 1) the route system (Interstate, Primary, or Secondary), 2) the Federal Aid route number, 3) a reference post on the route, and 4) the distance from that reference post to the milepoint being located.

To allow flexibility when stretches of highway are built, re-built, or abandoned, the reference posts are non-uniformly spaced. This means that the reference posts may be less than a mile apart, or may be more than a mile apart. A location 1.356 miles beyond reference post 45 (assuming reference post 46 is more than 1.356 miles from reference post 45) is referred to as 045+1.356.

A hypothetical stretch of highway is shown in Figure 1-II-1. This road is a portion of Federal Aid Primary Route 60, between reference posts 80 and 85. Milepoint "X" is located 0.378 miles beyond reference post 82. This milepoint may be uniquely specified by giving:

P	Primary Federal Aid route system,
060	Federal Aid Route number, and
082+0.378	Milepoint.

Roadlog records contain information about a section of highway rather than about a point on the highway. A "Roadlog section" is identified by specifying the beginning milepoint of the section. Hence, the Roadlog section between points X and Y in Figure 1-II-1 is identified by giving the milepoint of point X (P060082+0.378). Points X and Y are "section breaks" (a Roadlog record identified by milepoint X describes section X-Y, and another record identified by milepoint Y describes the next section). Section breaks occur at major junctions, city limits, county lines, and at any point at which the physical characteristics of the road (such as surface type or roadway width) change.

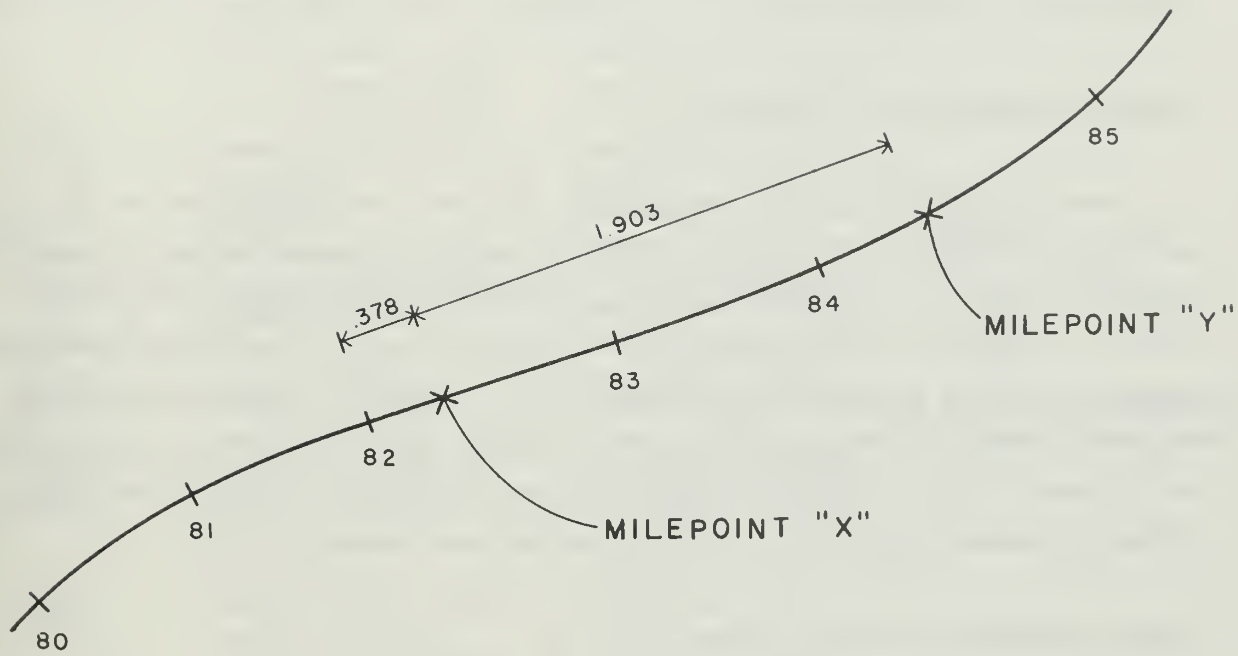


Figure 1-II-1. Hypothetical stretch of highway.

There are two major categories of records to be coded for initial construction of the Roadlog file: mileage records and descriptor records. Roadlog mileage and descriptor records are constructed on disk from data cards submitted by the user. This file construction is accomplished only when the files are first constructed. All file updating is accomplished through file maintenance which is explained in later sections of this chapter.

Roadlog Mileage Record Coding

Roadlog mileage records contain the physical and administrative data for sections of highway. The coding for each record is accomplished by preparing two data cards for each Roadlog mileage record in conformance with Table 1-II-I. Tables 1-II-II through 1-II-IX support Table 1-II-I (give additional details) and are referred to in Table 1-II-I. Roadlog mileage data may be differentiated from Roadlog descriptor data in that one of the following codes will appear in the "remarks" field (columns 65 and 66 of card 1) of Roadlog mileage records: blanks, NE, OS, SP, or LP (see Table 1-II-IV).

Roadlog Descriptor Record Coding

Actual physical and administrative data of Roadlog sections of highways are stored in the Roadlog mileage records. Roadlog descriptor records are necessary to the system in order to provide additional descriptions and allied information needed in the production of summaries that appear in the annual Montana Department of Highways Federal Aid Roadlog report.

Information in descriptor records may be valid for a Roadlog section, a number of sections, or for a milepoint. The five types of descriptor record information may be differentiated from Roadlog mileage record information by one of the following codes in the "remarks" field (columns 65 and 66 of card 1): DS, ER, EN, CO, or IL. The other difference, from a user's point of view, between Roadlog mileage record coding and Roadlog descriptor record coding is that there is only one card coded for descriptor records, as indicated in Table 1-II-X.

Table 1-II-X shows the formatting of the Roadlog descriptor record data and Table 1-II-XI explains the details of the five types of descriptor records. Tables 1-II-XII and 1-II-XIII support Table 1-II-XI, and are referred to in Table 1-II-XI.

TABLE 1-II-I

DATA ELEMENTS AND CODING INFORMATION
FOR ROADLOG MILEAGE RECORDS

Column(s)	Item	Remarks
	------------------------- First Data Card -------------------------	
1	Card Number	Code "1" in this field.
2	F. A. Route System	See Table 1-II-II.
3-5	F. A. Route Number	See Note 1 below.
6-8	Reference Post	See Note 1 below.
9	Plus Sign	Code "+" in this field.
10-14	Distance	See Note 2 below.
15-49	Description	Verbal description of section.
50-53	Maintenance Section	See Note 3 below.
54-64	Project Number	See Table 1-II-III.
65-66	Remarks	See Table 1-II-IV.
67-68	County Number	See Table 1-II-V.
69-70	Forest Highway	Forest highway number coded when applicable.
71-72	Administration Code	Coded, but not used in present subsystem.
73-76	First Location	See Table 1-II-VI.
77-80	Second Location	See Table 1-II-VI.
	------------------------- Second Data Card -------------------------	
1	Card Number	Code "2" in this field.
2-14	Route System, etc.	See Note 4 below.
15	Number of Lanes	Number of lanes must be coded.
16	Divided/Undivided	See Note 5 below.
17	Population	See Table 1-II-VII.
18-20	City Number	See Table 1-II-VIII.
21-22	Year Built	20th century assumed (e.g., 46=1946).
23-26	Surface Type	See Table 1-II-IX.
27-28	Surface Thickness	See Note 6 below.
29-31	Base Thickness	See Note 6 below.
32-33	Surface Width	See Note 7 below.
34-35	Roadway Width	See Note 7 below.
36-40	Route Length	See Note 8 below.
41-45	Constructed Length	See Note 9 below.
46-50	Unimproved Length	8.456 miles, code as 08456.
51-53	Wye Mileage	0.235 miles, code as 235.
54-58	Section Length	23.687 miles, code as 23687.
59-60	Year Improved	20th century assumed (e.g., 08=1908).
61-66	Date	Date from which record is applicable; January 3, 1972, code as 010372.
67-80	Unused Columns	

Notes:
1. Right-justified in field, code all leading zeroes.
2. Distance from last reference post to point on roadway of beginning of Roadlog section. Form is 9.999, code decimal point and all zeroes. Example: a point 0.368 miles beyond a particular reference post would be coded as 0.368.

3. Maintenance section numbers all contain four digits and a list of valid codes may be seen in the Montana Department of Highways 1971 Federal Aid Roadlog report.
4. Columns 2 thru 14 of card 2 are identical to columns 2 thru 14 of card 1.
5. All highways are either divided or undivided roadways. Code "D" for divided highways and "U" or leave blank for undivided highways.
6. Thickness to the tenth of an inch (e.g., 3.6 inch surface thickness, code as 36; 12.5 inch base thickness, code as 125; 6.5 inch base thickness code as 065).
7. Width in feet (e.g., 48' wide, code as 48).
8. All mileage records must contain a route length, which may range from 00.000 to 99.999 miles (86.957 miles, code as 86957; 2.345 miles, code as 02345). Route length is never smaller than section length, because center-line-to-shoulder lengths at intersections are included in route length. In most cases route length is identical to section length.
9. Constructed length is coded on all mileage records that end a project. Its value is the sum of the section lengths throughout the project, less any wye or unimproved mileage (8.623 miles, code as 08623).

TABLE 1-II-II

ROUTE SYSTEM CODES

Code*	Description
I	Federal Aid Interstate System.
P	Federal Aid Primary System.
S	Federal Aid Secondary System.

*These three codes are the only allowable codes until additional systems (e.g., the local system) are added to HIS.

TABLE 1-II-III

PROJECT NUMBER CODES

F	FI	FG	FGI
FU	S	SG	U
UI	UG	UGI	FL
FLI	FLG	FLGI	FHP
ERF	ERS	*CC	**MC
NFD	IN	USG	US
I	ING	DF	DFG
DS	DSG	DU	R-AD
IG	ERFO	CCC	DARM
ECHP	EFAP	EFHP	FAP
FAS	FAGH	FAGM	FAGS
FLP	NRFL	NRH	NRM
NRS	NP	WHP	WPMH
WPMS	WPSO	WPSS	WPGH
WPGM	WPGS	SC	SAP
WER	AE	UR	UF
WPH	A-AD		

*Printed as CNTY CONSTR in Federal Aid Roadlog Report.
**Printed as CITY CONSTR in Federal Aid Roadlog Report.

Note: The project number field consists of a left-justified project class code of from 1 to 4 characters in length, followed by a blank, and additional identification as appropriate. The project class code must be one of the classifications shown in this Table.

TABLE 1-II-IV

REMARK CODES

Code	Description	Type of Roadlog Record
NE	Non-existent mileage	Mileage record
OS	Out-of-state mileage	Mileage record
SP	Spur mileage	Mileage record
LP	Loop mileage	Mileage record
DS	Internal heading record	Descriptor record
ER	Additional description record	Descriptor record
EN	End-of-route record	Descriptor record
CO	Coincident mileage	Descriptor record
IL	Interstate Loop mileage	Descriptor record

Note: Most mileage records contain blanks in the remarks field. Only the remark codes listed above are allowable.

TABLE 1-II-V

COUNTY NAME CODES

Code	County Name	Code	County Name	Code	County Name
1	Beaverhead	20	Granite	38	Powder River
2	Big Horn	21	Hill	39	Powell
3	Blaine	22	Jefferson	40	Prairie
4	Broadwater	23	Judith Basin	41	Ravalli
5	Carbon	24	Lake	42	Richland
6	Carter	25	Lewis and Clark	43	Roosevelt
7	Cascade	26	Liberty	44	Rosebud
8	Chouteau	27	Lincoln	45	Sanders
9	Custer	28	McCone	46	Sheridan
10	Daniels	29	Madison	47	Silver Bow
11	Dawson	30	Meagher	48	Stillwater
12	Deer Lodge	31	Mineral	49	Sweet Grass
13	Fallon	32	Missoula	50	Teton
14	Fergus	33	Musselshell	51	Toole
15	Flathead	34	Park	52	Treasure
16	Gallatin	35	Petroleum	53	Valley
17	Garfield	36	Phillips	54	Wheatland
18	Glacier	37	Pondera	55	Wibaux
19	Golden Valley			56	Yellowstone

Note: The county number must be coded on all in-state mileage records (those without an "OS" code in remarks field). The county number field in out-of-state mileage records is left blank.

TABLE 1-II-VI

MILEAGE LOCATION CODES

Code	Description
CITY	City mileage
CNTY	County mileage
NFOR	National Forest mileage
IRES	Indian Reservation mileage
GAME	Game Refuge mileage
MRES	Military Reservation mileage
NMON	National Monument mileage
NPRK	National Park mileage
SFOR	State Forest mileage
SPRK	State Park mileage

Note: The first mileage location field for each Roadlog section (card 1, columns 73-76) must be coded on all Roadlog mileage records. The second mileage location field for each Roadlog section (card 1, columns 77-80) is zero (or blank) on most records, but contains a coded location when the section of roadway falls into two categories (e.g., municipal and National Forest).

TABLE 1-II-VII

POPULATION CODES

Code	Population of Incorporated City
1	0- 999
2	1,000- 2,499
3	2,500- 4,999
4	5,000- 9,999
5	10,000-24,999
6	25,000-49,999
7	50,000 and over

Note: A population code must be coded on all mileage records describing a section within an incorporated city, and must be zero (or blank) on all other records.

TABLE 1-II-VIII

CITY NAME CODES

Code	City	Code	City	Code	City
1	Alberton	43	Flaxville	85	Opheim
2	Anaconda	44	Forsyth	86	Outlook
3	Bainville	45	Fort Benton	87	Philipsburg
4	Baker	46	Froid	88	Plains
5	Bearcreek	47	Fromberg	89	Plentywood
6	Belgrade	48	Geraldine	90	Plevna
7	Belt	49	Glasgow	91	Polson
8	Big Sandy	50	Glendive	92	Poplar
9	Big Timber	51	Grassrange	93	Red Lodge
10	Billings	52	Great Falls	94	Rexford
11	Boulder	53	Hamilton	95	Richey
12	Bozeman	54	Hardin	96	Ronan
13	Bridger	55	Harlem	97	Roundup
14	Broadus	56	Harlowton	98	Ryegate
15	Broadview	57	Havre	99	Saco
16	Brockton	58	Helena	100	St. Ignatius
17	Browning	59	Hingham	101	Scobey
18	Butte	60	Hobson	102	Shelby
19	Cascade	61	Hot Springs	103	Sheridan
20	Chester	62	Hysham	104	Sidney
21	Chinook	63	Ismay	105	Stanford
22	Choteau	64	Joliet	106	Stevensville
23	Circle	65	Jordan	107	Sunburst
24	Clyde Park	66	Judith Gap	108	Superior
25	Columbia Falls	67	Kalispell	109	Terry
26	Columbus	68	Kevin	110	Thompson Falls
27	Conrad	69	Laurel	111	Three Forks
28	Culbertson	70	Lavina	112	Townsend
29	Cut Bank	71	Lewistown	113	Troy
30	Darby	72	Libby	114	Twin Bridges
31	Deer Lodge	73	Lima	115	Valier
32	Denton	74	Livingston	116	Virginia City
33	Dillon	75	Lodge Grass	117	Walkerville
34	Dodson	76	Malta	118	Westby
35	Drummond	77	Manhattan	119	West Yellowstone
36	Dutton	78	Medicine Lake	120	Whitefish
37	East Helena	79	Melstone	121	Whitehall
38	Ekalaka	80	Miles City	122	White Sulphur Springs
39	Ennis	81	Missoula	123	Wibaux
40	Eureka	82	Moore	124	Winifred
41	Fairfield	83	Nashua	125	Winnett
42	Fairview	84	Neihart	126	Wolf Point

Note: Valid city number codes are as listed above and must be coded on all mileage records describing a section within an incorporated city. All other records must contain zeroes (or blanks) in this field.

TABLE 1-II-IX

SURFACE TYPE CODES

Plant Mix Codes	Road Mix Codes	Primitive Code
4131	4134	0001
4154	4221	
4241	4231	
4251	4706	Unimproved Code
4252	9462	0002
4253	9662	
4254	9676	
4716	9740	Graded and Drained Codes
6101	9742	
6201		0010
6202		0011
6203	Portland Cement Concrete Codes	
6211		
6706	7001	Gravel Code
6805	7104	2010
6806	7201	
6807	7202	
	7204	Bituminous Surface Treatment Code
	7211	
	7212	3210
	7221	
	7222	
	7708	
	8301	

Note: Valid surface type codes are listed above and each Roadlog mileage record must have one, and only one, surface type coded in the field.

TABLE 1-II-X

DATA ELEMENTS AND CODING INFORMATION
FOR ROADLOG DESCRIPTOR RECORDS

Column(s)	Item	Remarks
- - - - - - - - - -	Only One Data Card	- - - - - - - - - -
1	Card Number	Code "1" in this field.
2	F. A. Route System	See Table 1-II-II.
3-5	F. A. Route Number	See Note 1 of Table 1-II-I.
6-8	Reference Post	See Note 1 of Table 1-II-I.
9	Plus Sign	Code "+" in this field.
10-14	Distance	See Note 2 of Table 1-II-I.
15-49	Description	See Table 1-II-XI.
50-64	Unused Columns	
65-66	Remarks	See Table 1-II-XI.
67-80	Unused Columns	

TABLE 1-II-XI

EXPLANATION OF DESCRIPTION FIELD OF ROADLOG DESCRIPTOR CARDS

If Remark in Columns 65 and 66 is:	Then Description Field (Columns 15-49) are Formatted:
EN	Unformatted description of the end of the route (e.g., JUNCTION FAP 20 NEAR SIDNEY) coded in the field. See Note 1 below.
ER	Unformatted description of the roadway (e.g., JUNCTION I 315 AT 10 AVE S INT) coded in the field. See Note 2 below.
DS	Unformatted descriptions of milepoint occurrences (e.g., US 12) coded in the field. See Note 3 below.
CO	See Table 1-II-XII.
IL	See Table 1-II-XIII.

Notes:

1. The "EN" code indicates the end of a route. A record of this type must be placed following the last Roadlog mileage record of each Federal Aid Route. EN records may not appear at any other locations in the Roadlog file.

2. The "ER" code indicates additional descriptive information needed in the description of the roadway in the annual Federal Aid Roadlog report.

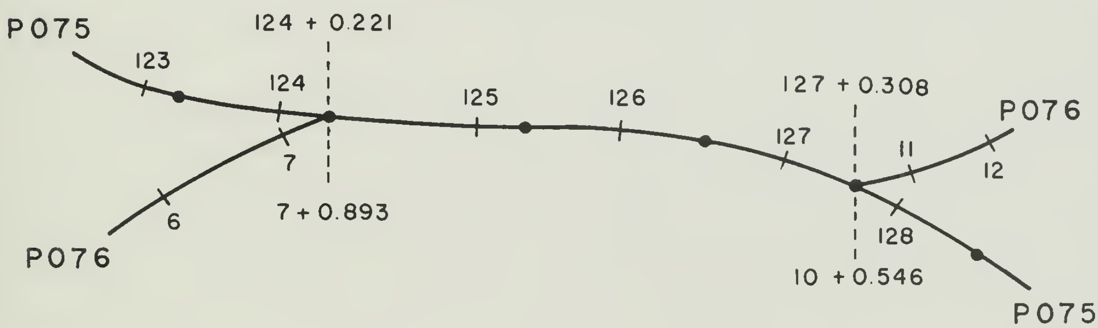
3. The "DS" code is currently used to indicate the signed route number, spur names, and loop names for succeeding road log entries. These "DS descriptions" are centered on the page of the Roadlog report with a blank line preceding and following the line of description. The first record of each route must be a "DS" record, indicating the signed route number. The milepoint field on this first record of each route must be left blank, to distinguish it from the first mileage record which is coded 000+0.000 in the milepoint field.

TABLE 1-II-XII

FORMAT OF DESCRIPTION FIELD FOR
"CO" DESCRIPTOR RECORDS

Card Column(s)	Coded Information
15-18	Code "COIN"
19	Must be left blank
20	F. A. Route System (I, P, or S)
21-23	F. A. Route Number (code leading zeroes)
24	Must be left blank
25-28	Code "FROM"
29	Must be left blank
30-38	Beginning milepoint of coincidence (e.g., 124+0.221), See Note below.
39	Code "-"
40-48	Ending milepoint of coincidence (e.g., 127+0.308), See Note below.
49	Must be left blank.

Note: "CO" records are used whenever two routes are coincident. The actual mileage records are coded on one route. A "CO" record is coded in the second route to direct the HIS routines to the location in the Roadlog file of the mileage records for the coincident section. Hence, the actual mileage records need not be re-coded for the second route. As an example of the use of a "CO" record, consider the following:



In this example, hypothetical routes 75 and 76 are coincident for several miles. Primary route 75 contains mileage records describing the sections of road on the coincident section. In record entries for Primary route 76, at milepoint 007+0.893 a Roadlog descriptor record is used to indicate the coincidence. The description field of this record must be coded in a fixed format (as shown in the Table above), giving the route number with which the present route is

TABLE 1-II-XII (continued)

coincident, and the milepoints of the beginning and ending points of the coincident section. In this case, the description is:

COIN P075 FROM 124+0.221-127+0.308

Only one "CO" record is used, irrespective of the number of Roadlog mileage records that are in the file for this same length of roadway on route 75. The milepoint coded for the next mileage record of route 76 is 010+0.546.

TABLE 1-II-XIII

FORMAT OF DESCRIPTION FIELD FOR "IL" DESCRIPTOR RECORDS

Column(s)	Coded Information
15-18	Code "LOOP"
19-49	Same as for "CO" card, see Table 1-II-XII and Note below.

Note: "IL" records are used in a fashion similar to "CO" records, but are used to describe "Interstate Loops." Interstate loops are completed sections of Interstate highway running parallel to Primary highway. The description has the same format as on "CO" records, with the keyword COIN replaced by LOOP. "IL" records are placed following the "EN" record ending a Primary route. One "IL" record must be included for each Interstate Loop occurring on the route. The route number coded in the description must be an interstate route. For example, suppose that Interstate route 15 runs parallel with Primary route 82 from milepoint 126+0.336 to 139+0.489 (milepoints on the Interstate route). An "IL" record is included following the "EN" record of route 82 containing the description:

LOOP I015 FROM 126+0.336-139+0.489

OS JCL Statements

Because HIS is designed to run in conjunction with the IBM System/360 Operating System, the Operating System must receive instructions from the user. Most of these instructions do not vary from run to run, and have been cataloged in a system library under the name HISRLG. The instructions in this procedure define to the Operating System the location of the data sets required by the Roadlog subsystem.

To utilize the Roadlog subsystem, the following OS JCL statements must be supplied:

```
// EXEC HISRLG,TIME=n
//SYSIN DD *
        Place HIS commands here.
/*
```

TIME=n gives the user the ability to specify the maximum time that he wants the computer to work on this particular run. "n" is the time in minutes.

HIS Commands

HIS command cards are identified by a colon (:) in column 1. All commands must contain this identification. Immediately following the colon is coded the name of the HIS program to be executed.

Most routines allow one or more options to be selected by the user. These options are selected by means of parameters coded on the command following the name of the program. Each parameter consists of a keyword and an option, separated by an equal sign (e.g., FILE=ROADLOG). The first parameter is separated by a comma from the program name. Additional parameters, if any, are separated by commas. The last parameter must be followed by at least one blank.

Continuation cards -- When a command is too large to be contained on a single card, it may be continued on another card by placing a comma after a

complete parameter, leaving the remainder of the card blank. The continuation card must contain a colon in column 1, followed by one or more blanks.

Comment cards -- Comments may be included in a group of commands by including a card with a "greater-than" sign (>) in column 1. Comment cards will be printed with the command listing, but are not processed.

Examples of valid HIS commands --

```
:LIST-&-SUM,REPORT=ROADLOG,DATA=PRIM,PAGE-NUMBER=60

:SURF-TYPE,REPORT=ROADLOG,DATA=SEC,SUMMARY=RTE-NO,
:          MILEAGE=ALL,TABLE-NUMBER=6

:UPDATE,FILE=ROADLOG,FUNCTION=INSERT,DDNAME=INDD
```

Examples of invalid HIS commands --

```
:UPDATE,FILE=ROADLOG,FUNCTION=INSERT,  DDNAME=INDD
```

(blank after third comma causes decoder to search for a continuation card, and DDNAME parameter is not recognized)

```
:LIST-&-SUM,REPORT=ROADLOG,DATA=PRIM,PAGE-NUMBER=60.
```

(final parameter must be followed by a blank)

```
*CREATE,FILE=ROADLOG
```

(first character must be a colon)

The DATA parameter -- Many HIS routines are set up to process a portion of the Roadlog file. The DATA parameter provides the user with a versatile method of indicating which Roadlog records are to be processed in producing a report or summary. A single route, a set of contiguous routes within a single route system, an entire route system, or the entire Federal Aid System may be specified. In order to retrieve an entire route system, DATA=INT, DATA=PRIM, or DATA=SEC is specified. The entire Federal Aid System is retrieved by specifying DATA=ALL. An additional form, DATA=INT+PRIM, is available to combine the Federal Aid Interstate and Primary systems. To retrieve a single route number, the route system is coded as above, followed by an equal sign and the route number. Leading zeroes need not be coded. For example,

DATA=INT=90, DATA=PRIM=3, and DATA=SEC=360 are valid DATA parameters for retrieving single routes. To retrieve a set of contiguous routes, a hyphen and a second route number are appended to the above form. For example, DATA=PRIM=2-5 causes Federal Aid Primary routes 2 through 5, inclusive, to be processed. One additional format of the DATA parameter is available, and may be used only on the SURF-TYPE command. This format is DATA=ILOOP, which causes Interstate Loops to be processed. In order to process a set of contiguous records in a file, the HIS routines require the keys of the first and last records. When the DATA parameter is used, the system calculates the appropriate beginning and ending keys. The user is relieved of this responsibility. It may occur, however, that a set of records are to be processed but cannot be specified through the DATA parameter. For instance, a portion of one route may need to be listed. The parameters STARTKEY and ENDKEY may be utilized to bypass the DATA parameter. Suppose that records 102+0.320 through 128+0.560 of Federal Aid Primary route 1 must be listed. Using the DATA parameter, the entire route would have to be processed. To bypass listing the entire route, DATA=PRIM=1 may be replaced by STARTKEY=P001102+0.320, ENDKEY=P001128+0.560. The keys are coded exactly as on Roadlog data cards. The STARTKEY and ENDKEY parameters may be used in lieu of the DATA parameter on any HIS command that allows specification of individual routes within the DATA parameters. They may not be used on those commands that allow only a combination of the DATA=INT, DATA=PRIM, DATA=INT+PRIM, DATA=SEC, or DATA=ALL forms of the DATA parameter.

<u>Report and report summary commands</u> -- The Roadlog report- and summary-generating programs provide the capability of producing most of the Montana Department of Highways annual Federal Aid Roadlog report. The Roadlog file, as well as several small subsidiary files, are utilized in producing these summaries. In the HIS commands the following are parametric definitions:

INT = Federal Aid Interstate System.
PRIM = Federal Aid Primary System.
SEC = Federal Aid Secondary System.
n = Federal Aid Route number for which report or summary is desired (e.g., 209).
n-n = Federal Aid Route numbers inclusively for which report or summary is desired (e.g., 209-211).

INT+PRIM = Federal Aid Interstate plus Federal Aid Primary Systems.
ALL = All of the Federal Aid Systems (Interstate + Primary + Secondary).
ILOOP = Interstate Loop system.
RTE-NO = Federal Aid Route number.
YR-BLT = Year built.
YR-IMP = Year improved.
SUR-WD = Surface width.
COUNTY = All Montana counties.
CITIES = All Montana cities.
PROJ-# = All project numbers.
URBAN = All Montana urban areas.
LOCATION = Location by jurisdictional boundary.

The summary-generating programs implemented in the Roadlog subsystem of HIS are:

<u>LIST-&-SUM COMMAND</u>:

$$\text{:LIST-\&-SUM,REPORT=ROADLOG,DATA=}\left\{\begin{array}{l}\text{INT}\\ \text{PRIM}\\ \text{SEC}\\[4pt] \text{INT=n}\\ \text{PRIM=n}\\ \text{SEC=n}\\[4pt] \text{INT=n-n}\\ \text{PRIM=n-n}\\ \text{SEC=n-n}\end{array}\right\}$$

LIST-&-SUM provides a formatted listing of Roadlog data, together with a summary by county location for each route. This listing forms the main body of the annual Federal Aid Roadlog report. Examples of LIST-&-SUM commands and accompanying OS JCL statements are:

```
// EXEC HISRLG,TIME=30
//SYSIN DD *
:LIST-&-SUM,REPORT=ROADLOG,DATA=PRIM
:LIST-&-SUM,REPORT=ROADLOG,DATA=SEC=201-300
:LIST-&-SUM,REPORT=ROADLOG,DATA=INT=94
/*
```

SURF-TYPE COMMAND:

```
                                             ⎧ INT          ⎫
                                             ⎪ INT+PRIM     ⎪
                                             ⎪ PRIM         ⎪
                                             ⎪ SEC          ⎪
                                             ⎪ ALL          ⎪
                                             ⎪ ILOOP        ⎪
:SURF-TYPE,REPORT=ROADLOG,DATA=              ⎨ INT=n        ⎬ ,
                                             ⎪ PRIM=n       ⎪
                                             ⎪ SEC=n        ⎪
                                             ⎪ INT=n-n      ⎪
                                             ⎪ PRIM=n-n     ⎪
                                             ⎩ SEC=n-n      ⎭

                 ⎧ RTE-NO ⎫
                 ⎪ YR-BLT ⎪
                 ⎪ YR-IMP ⎪
:     SUMMARY=   ⎨ SUR-WD ⎬    ,MILEAGE=  { URBAN }
                 ⎪ COUNTY ⎪               { ALL   }
                 ⎪ CITIES ⎪
                 ⎩ PROJ-# ⎭
```

A large number of summaries in the Roadlog report present a breakdown of section length according to surface type. These summaries are produced by SURF-TYPE. The mileage is shown according to one of the parameters route number, year built, year improved, surface width, county, city, or project class. The summaries may include only urban mileage, or urban and rural mileage. Examples of SURF-TYPE commands and accompanying OS JCL statements are:

```
// EXEC HISRLG,TIME=15
//SYSIN DD *
:SURF-TYPE,REPORT=ROADLOG,DATA=INT+PRIM,SUMMARY=YR-BLT,
:           MILEAGE=ALL
:SURF-TYPE,REPORT=ROADLOG,DATA=PRIM=6-20,
:       SUMMARY=COUNTY,MILEAGE=URBAN
:SURF-TYPE,REPORT=ROADLOG,DATA=ALL,SUMMARY=SUR-WD,MILEAGE=ALL
/*
```

SUMMARY-BY-ROUTES COMMAND:

```
                                            ⎧ INT      ⎫
                                            ⎪ INT+PRIM ⎪
:SUMMARY-BY-ROUTES,REPORT=ROADLOG,DATA=     ⎨ PRIM     ⎬
                                            ⎪ SEC      ⎪
                                            ⎩ ALL      ⎭
```

SUMMARY-BY-ROUTES provides a summary broken down by route number and by location code. Examples of SUMMARY-BY-ROUTES commands and accompanying OS JCL statements are:

```
// EXEC HISRLG,TIME=15
//SYSIN DD *
:SUMMARY-BY-ROUTES,REPORT=ROADLOG,DATA=INT+PRIM
:SUMMARY-BY-ROUTES,REPORT=ROADLOG,DATA=INT
:SUMMARY-BY-ROUTES,REPORT=ROADLOG,DATA=SEC
/*
```

SUMMARY-BY-LOCATION COMMAND:

```
                                            ⎧ INT      ⎫
:SUMMARY-BY-LOCATION,REPORT=ROADLOG,DATA=   ⎨ INT+PRIM ⎬
                                            ⎩ SEC      ⎭
```

SUMMARY-BY-LOCATION provides a summary broken down by route number and location (inside or outside of federal reservations). When DATA=INT+PRIM is specified, the status of the 7% system is also printed. The 7% system includes all mileage on the Interstate and Primary systems located outside federal reservations, excepting Interstate loops and urban extensions. The summary shows the total mileage outside federal reservations. From this figure is subtracted urban extension mileage. Non-urban Interstate loops are then subtracted (urban loops have already been subtracted as urban extensions). This result is the actual 7% system mileage, which is compared with the permissible amount of 4697.0 miles. Any underrun or overrun is printed. Examples of SUMMARY-BY-LOCATION commands and accompanying OS JCL statements are:

```
// EXEC HISRLG,TIME=10
//SYSIN DD *
:SUMMARY-BY-LOCATION,REPORT=ROADLOG,DATA=SEC
:SUMMARY-BY-LOCATION,REPORT=ROADLOG,DATA=INT
:SUMMARY-BY-LOCATION,REPORT=ROADLOG,DATA=INT+PRIM
/*
```

FORHWY-SUMMARY COMMAND:

$$:\text{FORHWY-SUMMARY,REPORT=ROADLOG,FHSUMMARY=} \left\{ \begin{matrix} \text{LOCATION} \\ \text{SURF-TYPE} \end{matrix} \right\}$$

FORHWY-SUMMARY provides two summaries for the Roadlog report. The first of these is a breakdown of forest highway mileage by location. The second is a breakdown of forest highway mileage by surface type. Examples of FORHWY-SUMMARY commands and accompanying OS JCL statements are:

```
// EXEC HISRLG,TIME=5
//SYSIN DD *
:FORHWY-SUMMARY,REPORT=ROADLOG,FHSUMMARY=SURF-TYPE
:FORHWY-SUMMARY,REPORT=ROADLOG,FHSUMMARY=LOCATION
/*
```

SUM-LOOPS-&-SPURS COMMAND:

```
:SUM-LOOPS-&-SPURS,REPORT=ROADLOG
```

This program provides a listing of Primary spurs and loops. Each spur/loop is printed on one line, showing the total route, constructed, unimproved, and wye mileages for the spur/loop. It is not possible to determine directly from the Roadlog file exactly when spurs and loops begin and end. Hence, the program must be supplied with data giving their locations. Each data card contains:

Columns 1-20:	Name of the spur/loop.
Columns 21-24:	Route system and number (Pnnn).
Column 25:	Blank.
Columns 26-34:	Beginning milepoint in form mmm+m.mmm.
Column 35:	Blank.
Columns 36-44:	End milepoint in form mmm+m.mmm.
Columns 45-80:	Not used--may contain any characters.

An example of the SUM-LOOPS-&-SPURS command and accompanying OS JCL statements is:

```
// EXEC HISRLG
//SPURTBL DD *
    data cards in above format
/*
//SYSIN DD *
:SUM-LOOPS-&-SPURS
/*
```

<u>Formatting option parameters</u> -- A number of formatting options are available under HIS to aid in the preparation of summaries for printing. These options are utilized by coding additional parameters which may be used on any HIS commands. Only those options applicable to the Roadlog report are presented in this section.

<u>FORMAT PARAMETER</u>:

FORMAT=REDUCE

The summaries printed for the Federal Aid Roadlog report require most or all of the 132 printer columns. After the computer printing, but before the reports are "commercially" printed, the pages are reduced. To indicate to HIS that reduction will be used, FORMAT=REDUCE may be coded on any individual command. The program SYS-PARAM may be used to set FORMAT=REDUCE into effect, alleviating the need to code it on each command. When FORMAT=REDUCE is coded, HIS will print the correct number of lines on each page, and position page numbers correctly. Examples of FORMAT=REDUCE parameters and accompanying OS JCL statements are:

```
// EXEC HISRLG,TIME=40
//SYSIN DD *
:LIST-&-SUM,REPORT=ROADLOG,DATA=INT,FORMAT=REDUCE
:LIST-&-SUM,REPORT=ROADLOG,DATA=PRIM,FORMAT=REDUCE
/*
```

```
// EXEC HISRLG,TIME=40
//SYSIN DD *
:SYS-PARAM,FORMAT=REDUCE
:LIST-&-SUM,REPORT=ROADLOG,DATA=INT
:LIST-&-SUM,REPORT=ROADLOG,DATA=PRIM
/*
```

PAGE-NUMBER PARAMETER:

$$\text{PAGE-NUMBER=} \left\{ \begin{array}{l} \text{n} \\ \text{\$+n} \\ \text{STOP} \end{array} \right\}$$

HIS does not begin numbering pages unless specifically requested to do so by the user. To begin numbering, specify PAGE-NUMBER=n on a command. The first page of output generated is numbered n, the next page is numbered n+1, and so on. Page numbering will continue until a command is encountered with a new PAGE-NUMBER parameter. If it is not known exactly how many pages will be produced, but it is known that several pages of maps, etc., are to be inserted between two pages, PAGE-NUMBER=$+n can be used to increment the current page number by n. Examples of PAGE-NUMBER parameters and accompanying OS JCL statements are:

```
// EXEC HISRLG,TIME=25
//SYSIN DD *
:SYS-PARAM,FORMAT=REDUCE
:SURF-TYPE,REPORT=ROADLOG,SUMMARY=RTE-NO,DATA=PRIM,MILEAGE=ALL
:SUMMARY-BY-LOCATION,REPORT=ROADLOG,DATA=INT+PRIM,
:             PAGE-NUMBER=305
:SUMMARY-BY-ROUTES,REPORT=ROADLOG,DATA=PRIM
:LIST-&-SUM,REPORT=ROADLOG,DATA=INT,PAGE-NUMBER=$+4
:LIST,FILE=ROADLOG,DATA=SEC=201-250,PAGE-NUMBER=STOP
/*
```

TABLE-NUMBER PARAMETER:

$$\text{TABLE-NUMBER=} \left\{ \begin{array}{l} \text{n} \\ \text{STOP} \end{array} \right\}$$

Many of the summaries in the Roadlog report are identified by a table number. By specifying TABLE-NUMBER=n on a command, HIS will

print the number of the table (n) at the top of each page produced by the command. Subsequent summaries will be numbered sequentially until the presence of a new TABLE-NUMBER parameter. By specifying TABLE-NUMBER=STOP on a command, HIS suspends the table numbering sequence. Examples of TABLE-NUMBER parameters and accompanying OS JCL statements are:

```
// EXEC HISRLG,TIME=14
//SYSIN DD *
:SYS-PARAM,FORMAT=REDUCE,PAGE-NUMBER=300
:SUMMARY-BY-LOCATION,REPORT=ROADLOG,DATA=INT+PRIM,
:                    TABLE-NUMBER=1
:SUMMARY-BY-LOCATION,REPORT=ROADLOG,DATA=SEC
:SURF-TYPE,REPORT=ROADLOG,SUMMARY=YR-BLT,
:          DATA=INT,MILEAGE=ALL,
:          TABLE-NUMBER=STOP
/*
```

TOP-MARGIN PARAMETER:

TOP-MARGIN=n

When this parameter is coded on a command, the number of blank lines (n) specified is padded at the top of the page. This parameter can be used to center output on a page when a summary requires less than one full page.

PAGE-EJECT PARAMETER:

PAGE-EJECT=SUPPRESS

This parameter is used to place output from a command on the same page as output from another command, and may be used in conjunction with TOP-MARGIN to place two small summaries on the same page, and centered on the page. For example, the Interstate SUMMARY-BY-ROUTES and SUMMARY-BY-LOCATION are two small summaries, each requiring only a few lines. These summaries may be placed on the same page and centered by the following commands:

```
// EXEC HISRLG,TIME=4
//SYSIN DD *
:SYS-PARAM,FORMAT=REDUCE
:SUMMARY-BY-ROUTES,REPORT=ROADLOG,DATA=INT,TOP-MARGIN=10
:SUMMARY-BY-LOCATION,REPORT=ROADLOG,
:            DATA=INT,PAGE-EJECT=SUPPRESS
/*
```

Roadlog file maintenance -- HIS is totally dependent on the data in its files. To aid the user in preparing accurate data, a set of file maintenance programs has been included within the system. These routines allow the user to update the file, obtain listings of the file, and to save backup copies of the file.

DUMP COMMAND:

$$\text{:DUMP,FILE=ROADLOG,DATA=} \left\{ \begin{array}{l} \text{INT} \\ \text{PRIM} \\ \text{SEC} \\ \text{INT=n} \\ \text{PRIM=n} \\ \text{SEC=n} \\ \text{INT=n-n} \\ \text{PRIM=n-n} \\ \text{SEC=n-n} \end{array} \right\}$$

DUMP provides an unformatted "dump" listing of the Roadlog file over the specified range of data. Examples of DUMP commands and accompanying OS JCL statements are:

```
// EXEC HISRLG,TIME=5
//SYSIN DD *
:DUMP,FILE=ROADLOG,DATA=INT
:DUMP,FILE=ROADLOG,DATA=SEC=209
:DUMP,FILE=ROADLOG,DATA=PRIM=3-6
/*
```

LIST COMMAND:

```
                                 ⎧ INT      ⎫
                                 ⎪ PRIM     ⎪
                                 ⎪ SEC      ⎪
                                 ⎪          ⎪
                                 ⎪ INT=n    ⎪
:LIST,FILE=ROADLOG,DATA=        ⎨ PRIM=n   ⎬
                                 ⎪ SEC=n    ⎪
                                 ⎪          ⎪
                                 ⎪ INT=n-n  ⎪
                                 ⎪ PRIM=n-n ⎪
                                 ⎩ SEC=n-n  ⎭
```

LIST provides a formatted listing of the Roadlog file over the specified range of data. In order to make the listing more easily read, only selected fields are printed. The fields printed are:

Key	Route Length
Description	Section Length
Project Class	Constructed Length
Year Built	Unimproved Length
Surface Type	Wye Length
Surface Width	Remark
Roadway Width	County Number
Number of Lanes	Population
Divided/Undivided	City Number
Location Codes	

In addition, the section length is accumulated from the beginning of the route, and this accumulated value printed in each record. Examples of LIST commands and accompanying OS JCL statements are:

```
// EXEC HISRLG,TIME=10
//SYSIN DD *
:LIST,FILE=ROADLOG,DATA=PRIM
:LIST,FILE=ROADLOG,DATA=INT=90
:LIST,FILE=ROADLOG,DATA=SEC=201-209
/*
```

LIST-ILOOPS COMMAND:

```
:LIST-ILOOPS,FILE=ROADLOG
```

LIST-ILOOPS has been supplied to aid in the coding of Interstate loops. The program searches the Primary portion of the file for "IL" records defining Interstate loops. Each time an "IL" record is found, the keys defining the loop are formed from the description, and the records comprising the loop are listed. The length of the loop is also printed. After all the loops have been listed, the total Interstate loop mileage is printed. An example of a LIST-ILOOPS command and accompanying OS JCL statements is:

```
// EXEC HISRLG,TIME=5
//SYSIN DD *
:LIST-ILOOPS,FILE=ROADLOG
/*
```

UPDATE COMMANDS:

```
                                      ⎧ DELETE  ⎫
:UPDATE,FILE=ROADLOG,FUNCTION=        ⎨ INSERT  ⎬ ,DDNAME=ddname
                                      ⎪ REWRITE ⎪
                                      ⎩ NEW-KEY ⎭
```

In addition to the UPDATE command necessary to invoke the proper update routine, the user must supply data describing the records to be updated. The DDNAME parameter informs HIS of the DD statement used to supply this data. When deleting a record, the DELETE option is specified and only the key is required. One data card is coded for each record being deleted, and contains the key in columns 1-13 of the card. The remainder of the card is blank. The record corresponding to the key coded will be deleted from the file. If no record with the specified key exists in the file, an error message is printed. When a new record is to be inserted into the file, the INSERT option is specified and fields must be supplied. A two-card sequence, consisting of a "1" card followed by a "2" card, is required to provide this information. The formats of these cards are shown in Tables 1-II-I. Whenever a mileage record is to be inserted, both cards are required. However, when descriptor records are inserted, all the information may be coded on a "1" card, and

the "2" card need not be supplied. If a record already exists in the file with the key coded, an error message is printed, and the record is not inserted into the file. When rewriting a record that already exists in the file, the REWRITE option is specified and only those fields which require alteration need to be coded. All other fields will remain unchanged. Any field but the key field may be changed in this manner. The same card formats are used when rewriting as when inserting. However, since only those fields being altered are coded, either a "1" card, a "2" card, or both may be coded. If a field is to be filled with blanks, the field is coded as filled with dollar signs ($). Because the rewrite function cannot alter the key field, the FUNCTION=NEW-KEY option must be coded when a key is in error and must be altered. The data cards for this function contain the existing key in columns 1-13, and the key to be substituted in columns 15-27. Columns 14 and 28-80 are blank. In order to alter the key, the existing record is deleted from the file, and inserted with a new key. If either: 1) a record already exists with the new key, or 2) no record exists with the old key, an error message is printed, and no change made to the file. An example of UPDATE commands and accompanying OS JCL statements is:

```
// EXEC HISRLG,TIME=5
//SYSIN DD *
:UPDATE,FILE=ROADLOG,FUNCTION=INSERT,DDNAME=ABCDE
:UPDATE,FILE=ROADLOG,FUNCTION=REWRITE,DDNAME=MYDATA
/*
//ABCDE DD *
    insertion data
/*
//MYDATA DD *
    rewrite data
/*
```

COPY COMMAND:

```
:COPY,FILE=ROADLOG,LIST= { YES }
                         { NO  }
```

The entire permanent file is copied into a backup area (tape or disk). An optional listing of the file in "dump" format may be obtained. Because the backup file may be placed on a tape or disk chosen by the user, the backup area must be defined to the IBM Operating System. This is done by means of a DD statement named SAVERLG. A complete discussion of OS Job Control Language for defining the backup area is beyond the scope of this manual; refer to the IBM publication OS Job Control Language Reference Manual. An example of a COPY command and accompanying OS JCL statements is:

```
// EXEC HISRLG,TIME=5
//SAVERLG DD UNIT=TAPE,VOL=SER=012345,DISP=(NEW,KEEP),
//           DSNAME=HIS.ROADLOG.BACKUP,
//           DCB=BLKSIZE=12000,LRECL=120,RECFM=FB)
//SYSIN DD *
:COPY,FILE=ROADLOG,LIST=YES
/*
```

<u>CREATE COMMAND</u>:

```
:CREATE,FILE=ROADLOG,LIST= { YES }
                           { NO  }
```

If the permanent file is destroyed, and a backup file (previously saved by COPY) exists on either tape or disk, CREATE may be used to restore the permanent file. A listing of the backup file (in "dump" format) is optional. As with COPY, the backup file reference must be defined by means of a DD statement named SAVERLG. An example of the CREATE command and accompanying OS JCL statements is:

```
// EXEC HISRLG,TIME=5
//SAVERLG DD UNIT=TAPE,VOL=SER=012345,DISP=OLD,
//           DSNAME=HIS.ROADLOG.BACKUP
//SYSIN DD *
:CREATE,FILE=ROADLOG,LIST=YES
/*
```

Storage of the Roadlog data -- All of the Roadlog Data is stored on a magnetic disk pack for access by the HIS System. As the data is transferred from the user's format shown in Table 1-II-I to magnetic disk, the formatting of many of the data fields are changed for more efficient storage and accessibility. For details regarding the internal format of the Roadlog Data see Highway Information System Volume 2: Programmer Information.

CHAPTER 1-III

TRAFFIC AND TRUE MILEAGE USER INFORMATION

The Traffic File

The Traffic file is a disk-resident file containing data for the three years prior to the current year. For each of these years is coded the ADT's at each point at which a count was taken or estimated, as well as the percentages of the ADT which are out-of-state vehicles, pickups, or commercial vehicles.

In addition to the complete three years, a fourth field is included for filling in data during the current year. At the end of the year, the fields are shifted one position, shifting the oldest data out of the record, and leaving the fourth field available for current-year data.

Data is stored according to Federal Aid routes. To uniquely specify a point on a highway, it is necessary to specify the route system (Interstate, Primary, or Secondary), route number, and milepoint. This information is used as a "key" in storing the data.

There are two major categories of records in the Traffic file: traffic count records and descriptor records. The record type is indicated through a 1-character "Remark" code within the record.

Traffic Count Records

"Traffic count" records are records containing annual average daily traffic (ADT) counts at specific milepoints of the Federal Aid system during each of four years. The data coded at a milepoint may be an actual or an estimated value. The coding of traffic count records is accomplished by preparing two data cards in accordance with Table 1-III-I. Table 1-III-II supports Table 1-III-I, and is referred to in Table 1-III-I.

Traffic count records are further subdivided into two subgroups: major section break records and minor section break records. Major section breaks occur at county lines, city limits, major junctions, and other locations at which breaks are desired in the Montana Department of Highways Federal Aid

TABLE 1-III-I

DATA ELEMENTS AND CODING INFORMATION
FOR TRAFFIC COUNT RECORDS

Column(s)	Item	Remarks
	- - - - - - - - - First Data Card - - - - - - - - -	
1	F. A. Route System	See Table 1-II-II.
2-4	F. A. Route Number	See Note 1 of Table 1-II-I.
5-7	Reference Post	See Note 1 of Table 1-II-I.
8	Plus Sign	Code "+" in this field.
9-13	Distance	See Note 2 of Table 1-II-I.
14	Actual/Estimated Code	See Note 1 below.
15-20	Unused Columns	
21-22	First (oldest) year	See Note 2 below.
23-27	ADT - first year	See Note 3 below.
28-30	Percentage of out-of-state vehicles - first year	See Note 4 below.
31-33	Percentage of pickups - first year	See Note 4 below.
34-36	Percentage of commercial vehicles - first year	See Note 4 below.
37-38	Second year	See Note 2 below.
39-43	ADT - second year	See Note 3 below.
44-46	Percentage of out-of-state vehicles - second year	See Note 4 below.
47-49	Percentage of pickups - second year	See Note 4 below.
50-52	Percentage of commercial vehicles - second year	See Note 4 below.
53-54	Third year	See Note 2 below.
55-59	ADT - third year	See Note 3 below.
60-62	Percentage of out-of-state vehicles - third year	See Note 4 below.
63-65	Percentage of pickups - third year	See Note 4 below.
66-68	Percentage of commercial vehicles - third year	See Note 4 below.
69-71	Future Factor	Coded but not used in present subsystem.
72-74	Design Hour Volume	See Note 4 below.
75	Remark	See Table 1-III-II.
76-80	Unused Columns	
	- - - - - - - - - Second Card - - - - - - - - -	
1	Continuation code	Code "C" in this field.
2-14	Key	See Note 5 below.
15-16	Fourth year	See Note 2 below.
17-21	ADT - fourth year	See Note 3 below.
22-24	Percentage of out-of-state vehicles - fourth year	See Note 4 below.

TABLE 1-III-I (continued)

Column(s)	Item	Remarks
	- - - - - - - - - Second Card Continued - - - - - - - - -	
25-27	Percentage of pickups - fourth year	See Note 4 below.
28-30	Percentage of commercial vehicles - fourth year	See Note 4 below.
31	Remark	See Note 6 below.
32-34	Design Hour Volume	See Note 6 below.
35-80	Unused columns	

Notes:
1. Code "A" for actual data and "E" for estimated data.
2. 20th century assumed (e.g., code "72" for 1972).
3. ADT is coded as right-justified 5-digit number. Leading zeroes must be included. For instance, code 346 as 00346.
4. Percentage is coded as 3-digit number (e.g., 42.3% is coded as 423).
5. Columns 2-14 of second card are identical to columns 1-13 of first card.
6. Space is available for the remark code and design hour volume on both the first and second cards. These may be coded on either card, or on both cards.

TABLE 1-III-II

REMARK CODES

Code	Description	Type of Traffic Record
W	Rural section	Major section break.
T	Municipal section	Major section break.
O	Out-of-state section	Major section break.
N	Non-existent section	Major section break.
R	Rural or out-of-state subsection	Minor section break.
M	Municipal subsection	Minor section break.
L	Loop	Descriptor
S	Spur	Descriptor
C	Coincident	Descriptor

Traffic by Sections report. Minor section breaks are locations within a traffic section at which traffic counts have been taken or estimated.

When printing the Traffic by Sections report, the description and county number are retrieved from the Roadlog file at each major section break. Hence, a Roadlog mileage or descriptor record (with an identical key) must exist for each major section break record in the Traffic file.

Data coded at a minor section break is used in weighting the ADT of its section, but no break is printed in the Traffic by Sections at the minor section break. For this reason, minor section break records need not correspond to Roadlog records.

Traffic Descriptor Records

Traffic descriptor records are used to identify the beginnings of spurs, loops, and coincident sections and the three remark codes for these records are "S," "L," and "C," respectively (see Table 1-III-II). No traffic data is coded on these records. A corresponding Roadlog descriptor record (with an identical key) must exist for each Traffic descriptor record. In the case of "C" records, the corresponding Roadlog record must be a "CO" type record defining a coincident section. In the case of "S" or "L" records, the corresponding Roadlog record must be a "DS" type record naming the spur or loop.

The format of Traffic descriptor records is shown in Table 1-III-III.

Coding Traffic Sections

Defining traffic sections -- Since each record in the file contains data for a point only, at least two records are necessary to define a section. The first record of a section is a major section break record (W, T, O, or N) defining the type of section. The last record of a section is another major section break record. If another section of road contiguous to the present section exists, this record is coded to indicate the type of section which follows. If no contiguous section follows (i.e., the end of the route has been reached, a coincident section follows, or a loop or spur begins), the last record of the section should contain the same code as the beginning

TABLE 1-III-III

DATA ELEMENTS AND CODING INFORMATION
FOR TRAFFIC DESCRIPTOR RECORDS

Column(s)	Item	Remarks
- - - - - - - - -	Only One Data Card	- - - - - - - - -
1-13	Key	See Note below.
14-74	Unused columns	
75	Remark	See Table 1-III-II
76-80	Unused columns	

Note: Columns 1-13 of Traffic descriptor data cards are identical to columns 1-13 of the first card of Traffic Section Break Records as shown in Table 1-III-I.

record. If the section is not a non-existent section (which has no traffic counts), as many sub-section records as required may be included between the beginning and ending major section break records. If a roadway discontinuity occurs (as described above), the following conventions must be followed:

1. End of Route.

   The last section of the route must contain the same major section break code at the beginning and end of the section. No other indication is necessary, as the next record in the file will contain a different route number.

2. Beginning of Coincident Section.

   When a route is coincident with a lower-number route, no traffic counts are taken for the higher-number route. A dummy C record is coded at the point where the coincident section begins. A major section break record, ending the previous section, is coded .01 mile or less upstream from the C record.

3. Beginning of Spur or Loop.

   Although reference posted contiguously with the main portion of the route, each spur/loop beginning represents a discontinuity in the highway. The previous section is terminated with the proper section break code. An "S"/"L" dummy record follows this record; the Roadlog file must contain a corresponding description. The dummy record cannot actually begin a new section; the proper major section break code must appear on the first record of the section.

Examples of highway sections are:

Rural Section

W
R
R
W

Municipal Section

T
M
T

Non-existent section

N
N

Two rural sections preceding a coincident section; coincident section followed by another rural section

W
R
W
R
R
R
W
C
W
R
R
W

$\left.\begin{matrix} W \\ C \end{matrix}\right\}$ .01 mile or less

Municipal section following rural section

W
R
T
M
M
T

Non-existent section preceded and followed by rural section

W
R
N
W
R
R
W

Rural loop following the main portion of a route

W
R
R
W
L
W
R
R
R
W

<u>Rural and out-of-state sections</u> -- The Traffic file contains room for four data years. Three positions are utilized to contain data for the three most recent years. The fourth is used to update the file during the year as

counts are taken. Once a year these values are shifted. Normally, rural sections contain data counts for all three of the retained years. Counting substations are rarely abandoned; when this occurs, the corresponding record is deleted from the file. In the case of a newly-constructed road, however, three years will pass before all three fields will contain data. When a new road is built, the records in the file corresponding to the old section are deleted. New records with data for the new road for the latest year only are coded. When producing the Traffic-by-Sections report, the ADT's for the earlier years are computed only on the basis of coded data. If no data at all is coded within a section, no values are computed. If data is given at the beginning and end of a section (i.e., the new road comprises only a portion of the section), the values are computed and weighted as if the uncoded records were not present. (Note that these records are used for computing the values for the most recent year.) Records within Rural and Out-of-state sections always contain values for the most recent year and contain values for the other years if the road has been in existence long enough.

Municipal sections -- Traffic counts in municipal sections, unlike those for rural sections, need not be taken every year. Hence, only years in which data is actually taken need be coded. Uncoded data anywhere in a section for a given year will cause the Traffic-by-Sections program to skip over this section in calculating ADT's for that year. When a T record terminates a rural or out-of-state section, it must be coded in the same manner as a W record.

Non-existent sections -- No Traffic counts are within non-existent sections. Hence, N records need not contain data unless they are used to terminate another section. For example, in the sequence:

W
R
N
N
W
R
R
W

the first N record terminates the first rural section. Data is coded on this record to allow computation on the rural section. The second N record

terminates the first non-existent section, and begins another. No data will be coded on this record. The W following this N terminates the second non-existent section, and begins a rural section. This record, of course, must contain data for this rural section.

The True Mileage File

The True Mileage file is a disk-resident file. There is one record in the file for each reference post on the Federal Aid Interstate, Primary and Secondary systems. The data in the file gives the location of each reference post relative to the beginning of the route on which it is located. The format of True mileage data cards is shown in Table 1-III-IV.

OS JCL Statements

Because HIS is designed to run in conjunction with the IBM System/360 Operating System, OS must receive instructions from the user. Most of these instructions do not vary from run to run, and have been cataloged in a system library under the name HISTRAF. The instructions in this procedure define to OS the locations of the data sets required by the Traffic and True Mileage subsystem.

To utilize the Traffic subsystem, the following OS JCL statements must be supplied:

```
// EXEC HISTRAF,TIME=n
//SYSIN DD *
    place HIS commands here
/*
```

HIS Commands

For HIS commands discussion, including continuation and comment cards, see Chapter 1-II.

Examples of valid HIS commands -- The following HIS commands and accompanying OS JCL statements are used in the production of the Montana

TABLE 1-III-IV

DATA ELEMENTS AND CODING INFORMATION
FOR TRUE MILEAGE RECORDS

Column(s)	Item	Remarks
1	F. A. Route System	See Table 1-II-II.
2-4	F. A. Route Number	See Note 1 of Table 1-II-I.
5-7	Reference Post	See Note 1 of Table 1-II-I.
8-13	True Mileage	See Note below.
14-80	Unused columns	

Note: The true mileage is the distance from the beginning of the route to the reference post. A decimal point is assumed between columns 10 and 11. For example, a true mileage of 21.361 is coded as 021361.

Department of Highways annual Traffic-by-Sections report:

```
// EXEC HISTRAF,TIME=20
//SYSIN DD *
:SYS-PARAM,FORMAT=NOREDUCE,PAGE-NUMBER=1
:TRAFFIC-BY-SECTIONS,REPORT=TRAFFIC,DATA=INT
:TRAFFIC-BY-SECTIONS,REPORT=TRAFFIC,DATA=PRIM
:TRAFFIC-BY-SECTIONS,REPORT=TRAFFIC,DATA=SEC
:SUMMARY-BY-ROUTES,REPORT=TRAFFIC,DATA=INT
:SUMMARY-BY-ROUTES,REPORT=TRAFFIC,DATA=PRIM
:SUMMARY-BY-ROUTES,REPORT=TRAFFIC,DATA=SEC
/*
```

Report and report summary commands -- The Traffic report and report summary commands provide the capability of producing the Montana Department of Highways annual Traffic-by-Sections report. A subsidiary file, the "Traffic Summary" file, is generated from the Traffic and True Mileage files. From this file, the Traffic-by-Sections report is printed.

CREATE-TRAFSUB COMMAND:

```
:CREATE-TRAFSUB
```

CREATE-TRAFSUB generates the Traffic Summary file, using the Traffic and True Mileage files. This file is subsequently utilized by the programs TRAFFIC-BY-SECTIONS and SUMMARY-BY-ROUTES to produce the Traffic-by-Sections annual report. An example of the CREATE-TRAFSUB command and accompanying OS JCL statements is:

```
// EXEC HISTRAF
//SYSIN DD *
:CREATE-TRAFSUB
/*
```

LIST-TRAFSUB COMMAND:

```
                       ⎧ INT       ⎫
                       ⎪ PRIM      ⎪
                       ⎪ SEC       ⎪
                       ⎪           ⎪
                       ⎪ INT=n     ⎪
:LIST-TRAFSUB,DATA=    ⎨ PRIM=n    ⎬
                       ⎪ SEC=n     ⎪
                       ⎪           ⎪
                       ⎪ INT=n-n   ⎪
                       ⎪ PRIM=n-n  ⎪
                       ⎩ SEC=n-n   ⎭
```

LIST-TRAFSUB provides a formatted listing of data in the Traffic Summary file. Examples of LIST-TRAFSUB commands and accompanying OS JCL statements are:

```
// EXEC HISTRAF
//SYSIN DD *
:LIST-TRAFSUB,DATA=INT
:LIST-TRAFSUB,DATA=PRIM=1-10
/*
```

TRAFFIC-BY-SECTIONS COMMAND:

```
                                           ⎧ INT       ⎫
                                           ⎪ PRIM      ⎪
                                           ⎪ SEC       ⎪
                                           ⎪           ⎪
                                           ⎪ INT=n     ⎪
:TRAFFIC-BY-SECTIONS,REPORT=TRAFFIC,DATA=  ⎨ PRIM=n    ⎬
                                           ⎪ SEC=n     ⎪
                                           ⎪           ⎪
                                           ⎪ INT=n-n   ⎪
                                           ⎪ PRIM=n-n  ⎪
                                           ⎩ SEC=n-n   ⎭
```

For each major section of Federal Aid Highway, the Traffic-by-Sections program prints a description of the section, the county location, the section length, the weighted average daily traffic for the latest three years, and the current vehicle miles (current year only). This summary forms the main body of the Traffic-by-Sections report of Federal Aid Highways. Examples of TRAFFIC-BY-SECTIONS commands and accompanying OS JCL statements are:

```
// EXEC HISTRAF
//SYSIN DD *
:TRAFFIC-BY-SECTIONS,REPORT=TRAFFIC,DATA=INT
:TRAFFIC-BY-SECTIONS,REPORT=TRAFFIC,DATA=PRIM=4
/*
```

SUMMARY-BY-ROUTES COMMAND:

$$\text{:SUMMARY-BY-ROUTES,REPORT=TRAFFIC,DATA=} \left\{ \begin{array}{l} \text{INT} \\ \text{PRIM} \\ \text{SEC} \end{array} \right\}$$

This program provides a summary of total vehicle miles, weighted ADT, and section length (rural highways only) for each route. Examples of SUMMARY-BY-ROUTES commands and accompanying OS JCL statements are:

```
// EXEC HISTRAF
//SYSIN DD *
:SUMMARY-BY-ROUTES,REPORT=TRAFFIC,DATA=INT
:SUMMARY-BY-ROUTES,REPORT=TRAFFIC,DATA=PRIM
:SUMMARY-BY-ROUTES,REPORT=TRAFFIC,DATA=SEC
/*
```

Formatting option parameters -- A number of formatting options are available under HIS to aid in the preparation of reports for printing. These options are utilized by coding additional parameters which may be used on any HIS commands. Only those options applicable to the Traffic-by-Sections report are discussed herein.

FORMAT=NOREDUCE PARAMETER:

In producing the Traffic-by-Sections report, the computer output is used directly, without reducing for printing. Specification of FORMAT=NOREDUCE sets the correct number of lines printed per page and sets the correct positioning of page numbers on the page. Program SYS-PARAM may be used to set FORMAT=NOREDUCE into effect for the duration of a system execution.

PAGE-NUMBER PARAMETER:

The PAGE-NUMBER parameter is discussed in Chapter 1-II.

Traffic file maintenance commands -- In order to aid the user in obtaining accurate results, a versatile set of file maintenance programs for the Traffic file are supplied. These programs are:

DUMP COMMAND:

$$\texttt{:DUMP,FILE=TRAFFIC,DATA=}\left\{\begin{array}{l}\texttt{INT}\\ \texttt{PRIM}\\ \texttt{SEC}\\[4pt] \texttt{INT=n}\\ \texttt{PRIM=n}\\ \texttt{SEC=n}\\[4pt] \texttt{INT=n-n}\\ \texttt{PRIM=n-n}\\ \texttt{SEC=n-n}\end{array}\right\}$$

DUMP provides an unformatted "dump" listing of the Traffic file over the specified range of data. Examples of DUMP commands and accompanying OS JCL statements are:

```
// EXEC HISTRAF,TIME=5
//SYSIN DD *
:DUMP,FILE=TRAFFIC,DATA=PRIM
:DUMP,FILE=TRAFFIC,DATA=INT=15
:DUMP,FILE=TRAFFIC,DATA=SEC=201-209
/*
```

LIST COMMAND:

$$\texttt{:LIST,FILE=TRAFFIC,DATA=}\left\{\begin{array}{l}\texttt{INT}\\ \texttt{PRIM}\\ \texttt{SEC}\\[4pt] \texttt{INT=n}\\ \texttt{PRIM=n}\\ \texttt{SEC=n}\\[4pt] \texttt{INT=n-n}\\ \texttt{PRIM=n-n}\\ \texttt{SEC=n-n}\end{array}\right\}$$

LIST provides a formatted listing of the file over the specified range of data. Only data for the three current years is printed; the fourth year (filled in during the year) can only be listed by DUMP. The key, remark code, true mileage, and Roadlog description is also printed in the listing. Examples of LIST commands and accompanying OS JCL statements are:

```
// EXEC HISTRAF
//SYSIN DD *
:LIST,FILE=TRAFFIC,DATA=INT
:LIST,FILE=TRAFFIC,DATA=PRIM=1-3
/*
```

UPDATE COMMAND:

```
                                   ⎧ DELETE  ⎫
:UPDATE,FILE=TRAFFIC,FUNCTION=     ⎨ INSERT  ⎬ ,DDNAME=ddname
                                   ⎨ REWRITE ⎬
                                   ⎩ NEW-KEY ⎭
```

In addition to the UPDATE command required to invoke the proper update routine, data must be supplied describing the records to be updated. The DDNAME parameter informs HIS of the DD statement used to supply this data. When deleting a record, the DELETE option is specified and only the key is required. One data card is coded for each record deleted; the card contains the key in columns 1-13 of the card. If no record exists in the file with the corresponding key, an error message is printed. Otherwise, the record is deleted. When a new record is inserted into the file, the INSERT option is specified and all applicable fields must be supplied. A two-card sequence is necessary if all four data fields are to be coded; in many cases, only one of these cards is required. Both cards allow the coding of a remark code and the design hour volume. If both cards are supplied, these values may be coded on either card (if coded on both, the values on the second card are used). The first card is used when the first three years are coded. The second card is used when the fourth year is coded. The card formats are shown in Table 1-III-I. If a record already exists in the file

with the key coded, an error message is printed, and the record is not inserted. When rewriting a record already existing in the file, the REWRITE option is specified and only those fields requiring alteration need to be coded; all other fields will remain unchanged. Any field but the key field may be altered with this function. The same card formats are used when rewriting as when inserting. If a field is to be filled with blanks, the field is coded as filled with dollar signs ($). The dollar signs are replaced by blanks prior to rewriting. If no record with the corresponding key exists in the file, an error message is printed. Because the rewrite function cannot alter the key field, the NEW-KEY option must be used when a key is changed. The data cards for this function contain the existing key in columns 1-13, and the key to be substituted in columns 15-27. The existing record is deleted from the file, and a new record (with the same information) is inserted with the new key. If either 1) a record already exists with the new key, or 2) no record exists with the old key, an error message is printed, the file remains unchanged. Examples of UPDATE commands and accompanying OS JCL statements are:

```
// EXEC HISTRAF
//SYSIN DD *
:UPDATE,FILE=TRAFFIC,FUNCTION=DELETE,DDNAME=WIPEOUT
:UPDATE,FILE=TRAFFIC,FUNCTION=INSERT,DDNAME=NEW
:UPDATE,FILE=TRAFFIC,FUNCTION=REWRITE,DDNAME=QRZ
/*
//WIPEOUT DD *

    data for deletion of records

/*
//NEW DD *

    data for insertion of records

/*
//QRZ DD *

    data for revision of records

/*
```

COPY COMMAND:

:COPY,FILE=TRAFFIC,LIST= $\left\{\begin{array}{c}\text{YES}\\ \text{NO}\end{array}\right\}$

The entire permanent file is copied into a backup area. An optional listing of the file in "dump" format may be obtained. The user must define the backup area (a tape or disk sequential file) by means of DD statement SAVETRF. A complete discussion of OS Job Control Language is beyond the scope of this manual; refer to the IBM publication OS Job Control Language Reference manual. An example of the COPY command with accompanying OS JCL statements is:

```
// EXEC HISTRAF
//SAVETRAF DD UNIT=TAPE,VOL=SER=111111,DISP=(NEW,KEEP),
//             DSNAME=HIS.TRAFFIC.BACKUP,LABEL=(1,SL,RETPD=365),
//             DCB=(BLKSIZE=3600,LRECL=80,RECFM=FB)
//SYSIN DD *
:COPY,FILE=TRAFFIC,LIST=NO
/*
```

CREATE COMMAND:

:CREATE,FILE=TRAFFIC,LIST= $\left\{\begin{array}{c}\text{YES}\\ \text{NO}\end{array}\right\}$

If the permanent file is destroyed, and a backup file (previously saved by COPY) exists on either tape or disk, CREATE may be used to restore the file. A list of the backup file (in "dump" format) is optional. As with COPY, the backup file reference must be defined by means of DD statement SAVETRF. An example of the CREATE command with accompanying OS JCL statements is:

```
// EXEC HISTRAF
//SAVETRF DD UNIT=TAPE,VOL=SER=111111,DISP=OLD,LABEL=(1,SL),
//            DSNAME=HIS.TRAFFIC.BACKUP
//SYSIN DD *
:CREATE,FILE=TRAFFIC,LIST=YES
/*
```

UPDATE-BY-YEAR COMMAND:

```
:UPDATE-BY-YEAR,FILE=TRAFFIC
```

UPDATE-BY-YEAR shifts the data years one position, clearing the fourth field for filling in with data during the upcoming year. For example, data for 1971 has been completed in the fourth field; the other three fields contain data for 1968, 1969, and 1970. When the UPDATE-BY-YEAR command is invoked, the 1968 data is disposed of, and the three fields will contain data for 1969 through 1971. The fourth field will contain zeroes and will be available to be filled in with 1972 data.

True Mileage file maintenance commands -- As in all other files it is important that the True Mileage file is kept accurate and up-to-date. The programs included in the True Mileage subsystem for maintaining the True Mileage file are:

LIST COMMAND:

$$\text{:LIST,FILE=TRUMILE,DATA=} \left\{ \begin{array}{l} \text{INT} \\ \text{PRIM} \\ \text{SEC} \\ \\ \text{INT=n} \\ \text{PRIM=n} \\ \text{SEC=n} \\ \\ \text{INT=n-n} \\ \text{PRIM=n-n} \\ \text{SEC=n-n} \end{array} \right\}$$

Examples of LIST commands and accompanying OS JCL statements are:

```
// EXEC HISTRAF
//SYSIN DD *
:LIST,FILE=TRUMILE,DATA=INT
:LIST,FILE=TRUMILE,DATA=PRIM=8-23
:LIST,FILE=TRUMILE,DATA=SEC=201
/*
```

UPDATE COMMAND:

$$:\text{UPDATE,FILE=TRUMILE,FUNCTION=} \left\{ \begin{array}{l} \text{DELETE} \\ \text{REWRITE} \\ \text{INSERT} \end{array} \right\} \text{,DDNAME=ddname}$$

When deleting a record, the DELETE option is specified and only the key needs to be given. One data card is supplied for each record deleted; the key is coded in columns 1-7 of the card. To rewrite one record, the REWRITE option is specified, the key is coded in columns 1-7 of a card, and the true mileage is coded in columns 8-13 (with a decimal assumed between columns 10 and 11). Hence, to rewrite the true mileage at reference post 389 on Federal Aid Primary route 75, which is located 391.046 miles from the beginning of the route, code:

P075389391046

It may happen that, due to a change in roadway, a number of reference post locations may change by a constant value. A second format of a rewrite card may be used to add or subtract a value between 0 and 99.999 to a set of consecutive records. The card format is:

Card Column	Contents
1-7	Adjustment factor in form ±nn.nnn (code leading zeroes).
8	Comma.
9	F. A. Route system (I, P, or S).
10-12	F. A. Route Number.
13	Comma.
14-16	Beginning reference post.
17-20	Comma and ending reference post (Optional).

The program reads the record corresponding to the beginning reference post, and adds the adjustment factor to the existing true mileage at that reference post. The record is then rewritten with the adjusted true mileage. The program continues sequentially through the file, adding the adjustment factor to subsequent records. If no ending reference post is coded (columns 17-20 are blank), the

process continues to the end of the route. If an ending reference post is present, the process continues through the reference post. Examples of REWRITE commands for adjusting a series of reference post locations are:

+00.203,P020,034	(.203 is added to all records from reference post 34 to the end of route 20).
-01.557,I090,124,148	(1.557 is subtracted from all records between 124 and 148, inclusive).

To insert a record, the INSERT option is specified and the entire record is coded. (The key is coded in columns 1-7, and the true mileage is coded in columns 8-13 with a decimal assumed between columns 10 and 11). Examples of UPDATE commands and accompanying OS JCL statements are:

```
// EXEC HISTRAF
//SYSIN DD *
:UPDATE,FILE=TRUMILE,FUNCTION=INSERT,DDNAME=GOTOIT
:UPDATE,FILE=TRUMILE,FUNCTION=REWRITE,DDNAME=XXXXX
/*
//GOTOIT DD *
    insert data

/*
//XXXXX DD *
    rewrite data

/*
```

COPY COMMAND:

```
:COPY,FILE=TRUMILE
```

DD statement SAVETRM must be supplied to define a sequential file on tape or disk in which the file is to be saved. An example of the COPY command and accompanying OS JCL statements is:

```
// EXEC HISTRAF
//SAVETRM DD UNIT=TAPE,VOL=SER=111111,DISP=(NEW,KEEP),
//            DSNAME=HIS.TRUMILE.BACKUP,LABEL=(2,SL,RETPD=365),
//            DCB=(BLKSIZE=3600,LRECL=16,RECRM=FB)
//SYSIN DD *
:COPY,FILE=TRUMILE
/*
```

CREATE COMMAND:

```
:CREATE,FILE=TRUMILE
```

CREATE is used when the permanent file is destroyed, and a backup copy has been saved by COPY. DD statement SAVETRM defines the backup file. An example of the CREATE command and accompanying OS JCL statements is:

```
// EXEC HISTRAF
//SAVETRM DD UNIT=TAPE,VOL=SER=111111,DISP=OLD,LABEL=(2,SL),
//            DSN=HIS.TRUMILE.BACKUP
//SYSIN DD *
:CREATE,FILE=TRUMILE
/*
```

Storage of the Traffic and True Mileage data -- All of the Traffic and True Mileage data is stored on a magnetic disk pack for access by the HIS System. As the data is transferred from the user's coding formats shown in Tables 1-III-I and 1-III-IV to magnetic disk, the formatting of many of the data fields are changed for more efficient storage and accessibility. For details regarding the internal format of the Traffic and True Mileage data see Highway Information System Volume 2: Programmer Information.

CHAPTER 1-IV

ACCIDENT USER INFORMATION

The Accident Files

Accident data is stored in two disk-resident indexed-sequential files. The first of these, the "Accident Detail" file, contains information pertaining to each accident as a whole, such as the date, time, location, etc. The second, the "Accident Vehicle" file, contains information pertaining to the vehicles and pedestrians involved in the accidents.

For a more detailed discussion on Accident data coding, refer to the publication "A Manual Describing the Proper Use of the State of Montana Investigator's Accident Report."

Accident Detail Record Coding

One Accident Detail record must be coded for each accident entered into the files. This record requires two data cards, the first referred to as an "A" card and the second as a "B" card. The formats of these cards are shown in Table 1-IV-I. Tables 1-IV-II through 1-IV-XVI support Table 1-IV-I, and are referred to in Table 1-IV-I.

Accident Vehicle Record Coding

One Accident Vehicle record must be coded for each vehicle and for each pedestrian involved in an accident entered into the files. Two data cards, a "C" card and a "D" card, are required for a vehicle record. Additional records, requiring only a "D" card, are coded for vehicles involving injuries to passengers within the vehicle other than in the positions front center, front right, rear left, rear center, or rear right, such as in the box of a pickup or in a bus. In this case, the single card is referred to as an "I" card rather than a "D" card.

The formats of "C," "D" and "I" cards are shown in Table 1-IV-XVII. Tables 1-IV-XVIII through 1-IV-XXII support Table 1-IV-XVII, and are referred to in Table 1-IV-XVII.

TABLE 1-IV-I

DATA ELEMENTS AND CODING INFORMATION
FOR ACCIDENT DETAIL RECORDS

Column(s)	Item	Remarks
- - - - - - - - -	"A" Card - - - - - - - - -	
1	Card Code	See Note 1 below.
2	Sequence Number	Always "0" on "A" cards.
3-14	Accident Number	See Note 2 below.
15-16	Month	See Note 3 below.
17-18	Day of Month	See Note 3 below.
19-20	Year	See Note 3 below.
21-22	Hour	24-Hour Clock.
23-24	Minute	
25-27	City Number	See Note 4 below.
28-29	County Number	See Note 5 below.
30-41	Accident Location	See Table 1-IV-III.
42-43	First Harmful Event	See Table 1-IV-IV.
44-45	First Object Hit Off Roadway	See Table 1-IV-V.
46	Injury Severity	See Table 1-IV-VI.
47	Damage Severity	See Table 1-IV-VII.
48	Class of Trafficway	See Table 1-IV-VIII.
49	Roadway-Related Location	See Note 6 below.
50	Junction-Related Location	See Table 1-IV-IX.
51-52	Number of Vehicles	See Note 7 below.
53-54	Number of Pedestrians	See Note 7 below.
55-56	Number of Fatalities	See Note 7 below.
57-58	Number of Injuries	See Note 7 below.
59	Weather Condition	See Table 1-IV-X.
60	Road Condition	See Table 1-IV-XI.
61	Light Condition	See Table 1-IV-XII.
62-63	Traffic Controls	See Table 1-IV-XIII.
64-65	Other Damage-Type	See Table 1-IV-XIV.
66	Other Damage-Severity	See Note 8 below.
67	Other Damage-Owner	See Note 9 below.
68-69	Posted Speed Limit	
70	Engineering Study	See Note 10 below.
71-72	Analysis-Field 1	See Table 1-IV-XV.
73-74	Analysis-Field 2	See Table 1-IV-XV.
75	Collision Type	See Table 1-IV-XVI.
76-80	Unused Columns	
- - - - - - - - -	"B" Card - - - - - - - - -	
1	"B"	See Note 11 below.
2	Sequence Number	Always "0" on "B" cards.
3-14	Accident Number	See Note 2 below.
15-24	Date and Time Notified	See Note 12 below.
25-34	Date and Time Arrived	See Note 12 below.
35-80	Unused Columns	

TABLE 1-IV-I (continued)

Notes:

1. Card code on "A" card is "A" for investigated accidents and "E" for uninvestigated accident.
2. The first two digits of the accident number must be the year in which the accident occurred (20th century assumed). The remaining 10 digits must be a unique number for each accident report consisting of a code for the agency filing the report and other identification making it a unique number.
3. The date and time coded in columns 15-24 are the date and time at which the accident occurred.
4. A city number is coded only when an accident occurs within the bounds of an incorporated city. The names of the incorporated cities, together with their corresponding city numbers, are shown in Table 1-II-VIII.
5. The county number is coded according to the licensing numbering system, rather than in the alphabetical number system used in the other HIS files. The county names and corresponding numbers are shown in Table 1-IV-II.
6. Code "1" for on roadway and "2" for off roadway.
7. Right-justified in field, code all leading zeroes.
8. Code "1" for minor, "2" for moderate, and "3" for major.
9. Code "1" for federal, "2" for state, "3" for county, "4" for city, and "5" for private.
10. Code an "X" in this column to request an engineering study.
11. "B" cards are never coded for uninvestigated accidents, and are optional for investigated accidents.
12. These columns are coded in the same format as columns 15-24 of the "A" card.

TABLE 1-IV-II

COUNTY NAMES AND NUMBERS
FOR ACCIDENT DETAIL FILE

Number	Name	Number	Name	Number	Name
1	Silver Bow	20	Valley	38	Glacier
2	Cascade	21	Toole	39	Fallon
3	Yellowstone	22	Big Horn	40	Sweet Grass
4	Missoula	23	Musselshell	41	McCone
5	Lewis and Clark	24	Blaine	42	Carter
6	Gallatin	25	Madison	43	Broadwater
7	Flathead	26	Pondera	44	Wheatland
8	Fergus	27	Richland	45	Prairie
9	Powder River	28	Powell	46	Granite
10	Carbon	29	Rosebud	47	Meagher
11	Phillips	30	Deer Lodge	48	Liberty
12	Hill	31	Teton	49	Park
13	Ravalli	32	Stillwater	50	Garfield
14	Custer	33	Treasure	51	Jefferson
15	Lake	34	Sheridan	52	Wibaux
16	Dawson	35	Sanders	53	Golden Valley
17	Roosevelt	36	Judith Basin	54	Mineral
18	Beaverhead	37	Daniels	55	Petroleum
19	Chouteau			56	Lincoln

TABLE 1-IV-III

ACCIDENT LOCATION FIELD

Column(s)	Contents	Remarks
	- - - - - - - - On-System Rural - - - - - - -	
30	F. A. Route System	See Table 1-II-II.
31-33	F. A. Route Number	See Note 1 below.
34-36	Reference Post	See Note 1 below.
37-41	Distance from Reference Post	See Note 2 below.
	- - - - - - - - Off-System Rural - - - - - - -	
30	"R"	
31	Blank	
32-33	Coordinates within Section	
34-36	Range	
37-39	Township	
40-41	Section	
	- - - - - - - - On-System Municipal - - - - - - -	
30	F. A. Route System	See Table 1-II-II.
31-33	F. A. Route Number	See Note 1 below.
34-37	X-Coordinate	From City Map.
38-41	Y-Coordinate	From City Map.
	- - - - - - - - Off-System Municipal - - - - - - -	
30	"M"	
31-33	City Number	See Table 1-II-VIII.
34-37	X-Coordinate	From City Map.
38-41	Y-Coordinate	From City Map.

Notes:
1. Right-justified in field, code all leading zeroes.
2. Code in form +nnnn. A decimal point is assumed after the first digit. For example, code a distance of 0.368 miles as +0368.

TABLE 1-IV-IV

FIRST HARMFUL EVENT CODES

Code	Event
01	Overturned.
02	Other Non-Collision.
03	Collision with Pedestrian.
04	Collision with Motor Vehicle in Transport.
05	Collision with Motor Vehicle in Other Roadway.
06	Collision with Parked Motor Vehicle.
07	Collision with Railway Train.
08	Collision with Pedalcycle.
09	Collision with Animal.
10	Collision with Fixed Object.
11	Collision with Other Object.

TABLE 1-IV-V

FIRST OBJECT HIT OFF ROADWAY CODES

Code	Object
01	End of Overpass or River Crossing.
02	Guardrail Protecting Overpass Structure.
03	Overpass Railing or Side of Overpass.
04	End of Underpass.
05	Pier of Underpass.
06	Guardrail Protecting Underpass.
07	Lighting, Power Pole, Signal Pole.
08	Guardrail Protecting Lighting or Power Pole.
09	Sign.
10	Guardrail Protecting Sign.
11	Median Guardrail.
12	Guardrail Along Fill.
13	End of Guardrail.
14	Other Guardrail.
15	Tree.
16	Cut Slope.
17	Road Approach.
18	Rock or Boulder.
19	End of Drainage Pipe.
20	Building or Other Structure.
21	Fence.
22	Raised Median or Curb.
23	Other Object.
24	No Object Hit (may be coded as 00).
25	Unknown.

TABLE 1-IV-VI

INJURY SEVERITY CODES

Code	Injury Severity
0	No Injury.
1	Fatal Injury.
2	Incapacitating Injury (Cannot Normally Perform).
3	Non-Incapacitating Injury (Evidence of Injury).
4	Possible Injury (Apparent Symptoms).

TABLE 1-IV-VII

DAMAGE SEVERITY CODES

Code	Damage Severity
0	No Damage.
1	Disabling Damage.
2	Functional Damage.
3	Other Motor Vehicle Damage.

TABLE 1-IV-VIII

CLASS OF TRAFFICWAY CODES

Code	Class of Trafficway
1	Interstate System
2	Other Fully Controlled Access Road
3	Other US Numbered Route
4	Other State Numbered Route
5	Other Major Arterial
6	County Road
7	Local Street
8	Other Road

TABLE 1-IV-IX

JUNCTION-RELATED LOCATION CODES

Code	Junction-Related Location
0	Non-Junction
1	Intersection
2	Intersection-Related
3	Driveway Access

TABLE 1-IV-X

WEATHER CONDITION CODES

Code	Weather Condition
1	Clear
2	Raining
3	Snowing
4	Fog
5	Other

TABLE 1-IV-XI

ROAD CONDITION CODES

Code	Road Condition
1	Dry
2	Wet
3	Snowy
4	Icy
5	Other

TABLE 1-IV-XII

LIGHT CONDITION CODES

Code	Light Condition
1	Daylight
2	Dawn or Dusk
3	Darkness, Lighted
4	Darkness, Unlighted
5	Other

TABLE 1-IV-XIII

TRAFFIC CONTROL CODES

Code	Traffic Controls
01	Traffic Signals
02	Traffic Signals not Working
03	Traffic Signals with Pedestrian Heads
04	Traffic Signals with Ped. Heads (Heads not Working)
05	Flasher
06	Flasher not Working
07	Stop Sign
08	Yield Sign
09	Railroad Signals
10	Railroad Signal not Working
11	Railroad Gates
12	Railroad Gates not Working
13	Do Not Enter Signs
14	Other Regulatory Sign
15	Warning Sign
16	Pavement Markings

TABLE 1-IV-XIV

OTHER DAMAGE TYPE CODES

Code	Other Damage Type
01	Signal, Lighting, Power Pole
02	Sign
03	Guardrail
04	Bridge
05	Building
06	Shrubbery/Trees
07	Maintenance Equipment
08	Fire Hydrant
09	Road Surface
10	Drainage Structure
11	Fence
12	Barricades
13	Other

TABLE 1-IV-XV

ANALYSIS CODES

Code	Analysis
01	Failed to have vehicle under control (speed not involved)
02	Inattentive driving
03	Inexperience
04	Blackout, heart, stroke, etc.
05	Fell asleep
06	Sun Glare
07	Raining
08	Snowing
09	Whiteout
10	Blowing Snow
11	Whiteout -- meeting or following vehicle
12	Dust Storm
13	Dust caused by wind or preceding vehicle on unoiled road surface
14	Road slippery or icy
15	Other weather conditions
16	Improper hitch
17	Blow out -- flat tire
18	Stone thrown by vehicle
19	Avoiding another vehicle
20	Avoiding pedestrian -- unexpected actions
21	Striking or avoiding domestic animal in roadway
22	Striking or avoiding wild animal in roadway
23	Striking or avoiding object in roadway
24	Distraction within vehicle
25	Distraction from outside vehicle
26	Unwarranted slowing
27	Blinded by glaring lights other than vehicle
28	Passenger fell from vehicle
29	Occupant releases vehicle
30	Indian in violation on reservation -- Patrol has no jurisdiction
31	Traffic control sign -- missing, down, etc.
32	Wind Blowing
33	Water on highway
34	Fog

TABLE 1-IV-XVI

COLLISION TYPE CODES

Code	Collision Type
1	Head On
2	Rear End
3	Angle
4	Sideswipe Meeting
5	Sideswipe Passing
6	Backed Into
7	Other

TABLE 1-IV-XVII

DATA ELEMENTS AND CODING INFORMATION
FOR ACCIDENT VEHICLE RECORDS

Column(s)	Item	Remarks
	- - - - - - - - - - "C" Card - - - - - - - - - -	
1	Card Code	See Note 1 below.
2	Sequence Number	
3-14	Accident Number	See Note 2 of Table 1-IV-I.
15-34	Last Name	See Note 2 below.
35	First Initial	
36	Middle Initial	
37-53	Driver's License Number	
54-55	State of Driver's License	See Table 1-IV-XVIII.
56-61	Date of Birth	See Note 3 below.
62	Re-Examination Code	See Note 4 below.
63-68	Charge Code	
69-74	Summons Number	
75-80	Unused Columns	
	- - - - - - - - - - "D" Card - - - - - - - - - -	
1	Card Code	See Note 5 below.
2	Sequence Number	
3-14	Accident Number	See Note 2 of Table 1-IV-I.
15-19	Contributing Circumstances	See Note 6 below.
20	Driver - Alcohol	See Note 7 below.
21-22	Driver - Age	
23	Driver - Sex	See Note 8 below.
24	Driver - Injury Severity	See Table 1-IV-VI.
25-29	Passenger - Front Center	See Note 9 below.
30-34	Passenger - Front Right	See Note 9 below.
35-39	Passenger - Rear Left	See Note 9 below.
40-44	Passenger - Rear Center	See Note 9 below.
45-49	Passenger - Rear Right	See Note 9 below.
50-51	Vehicle Number	See Note 10 below.
52-53	Vehicle Intent	See Table 1-IV-XX and Note 10 below.
54-55	Pedestrian Number	See Note 11 below.
56-57	Pedestrian Intent	See Table 1-IV-XX and Note 11 below.
58-59	Body Style	See Table 1-IV-XXI and Note 10 below.
60	Trailer Style	See Table 1-IV-XXII and Note 10 below.
61-62	Vehicle Year	See Note 10 below.
63	Vehicle Involved in Interstate Traffic	See Note 12 below.
64-78	Vehicle License Number or VIN	See Note 13 below.
79	Damage Severity	See Table 1-IV-VII.
80	Damage Level	See Note 14 below.
	- - - - - - - - - - "I" Card - - - - - - - - - -	
1	"I"	See Note 15 below.
2	Sequence Number	

TABLE 1-IV-XVII (continued)

Column(s)	Item	Remarks
- - - - - "I" Card (continued) - - - - -		
3-14	Accident Number	See Note 2 of Table 1-IV-I.
15-19	Unused Columns	
20-24	Additional Passenger	See Note 9 below.
25-29	Additional Passenger	See Note 9 below.
30-34	Additional Passenger	See Note 9 below.
35-39	Additional Passenger	See Note 9 below.
40-44	Additional Passenger	See Note 9 below.
45-49	Additional Passenger	See Note 9 below.
50-63	Unused Columns	
64-78	Vehicle License Number or VIN	See Note 16 below.
79-80	Unused Columns	

Notes:
1. Card code on "C" card is "C" for investigated accidents and "G" for uninvestigated accidents.
2. Name is left-justified and padded with blanks to complete field.
3. Birth date is coded as month in columns 56-57, day in columns 58-59, and year in columns 60-61.
4. Code an "X" to indicate re-examination.
5. Card code on "D" card is "D" for investigated accidents and "H" for uninvestigated accidents.
6. The contributing circumstances is made up of five separate 1-digit fields - vision, physical defects, road defects, mechanical defects, and possible violations. Table 1-IV-XIX shows the codes for each of these.
7. Code "0" for not drinking and "1" for had been drinking.
8. Code "M" for male and "F" for female.
9. Passenger information is in same format as driver information in columns 20-24 of "D" card.
10. If the vehicle record describes a vehicle code the information. Fill this field with zeroes if the record describes a pedestrian.
11. If the vehicle record describes a pedestrian code the information. Fill the field with zeroes if the record describes a vehicle.
12. Code an "X" if the vehicle was involved in Interstate traffic.
13. If the vehicle license number is available, code the number in columns 64-71, the state (see Table 1-IV-XVIII) in columns 72-73, and the license year in columns 74-75. Otherwise, code the vehicle identification number in columns 64-78.
14. Code an "X" if vehicle damage exceeds $250.
15. "I" cards are used to include additional injuries that cannot be coded on the "D" card for a vehicle. More than one "I" card may be included if necessary.
16. Identical to columns 64-78 of the "D" card. See note 13.

TABLE 1-IV-XVIII

STATE CODES

Code	State	Code	State
AL	Alabama	MT	Montana
AK	Alaska	NB	Nebraska
AZ	Arizona	NV	Nevada
AR	Arkansas	NH	New Hampshire
CA	California	NJ	New Jersey
CO	Colorado	NM	New Mexico
CT	Connecticut	NY	New York
DE	Delaware	NC	North Carolina
DC	District of Columbia	ND	North Dakota
FL	Florida	OH	Ohio
GA	Georgia	OK	Oklahoma
GU	Guam	OR	Oregon
HI	Hawaii	PA	Pennsylvania
ID	Idaho	PR	Puerto Rico
IL	Illinois	RI	Rhode Island
IN	Indiana	SC	South Carolina
IA	Iowa	SD	South Dakota
KS	Kansas	TN	Tennessee
KY	Kentucky	TX	Texas
LA	Louisiana	UT	Utah
ME	Maine	VT	Vermont
MD	Maryland	VA	Virginia
MA	Massachusetts	VI	Virgin Islands
MI	Michigan	WA	Washington
MN	Minnesota	WV	West Virginia
MS	Mississippi	WI	Wisconsin
MO	Missouri	WY	Wyoming

TABLE 1-IV-XIX

CONTRIBUTING CIRCUMSTANCES

Vision

0 Vision not obscured
1 Buildings
2 Trees or hedges
3 Other Vehicle
4 Smoke
5 Dust
6 Other

Mechanical Defects

0 No apparent mechanical defects
1 Lights
2 Brakes
3 Tires or steering
4 Other

Physical Defects

0 No apparent defects
1 Vision
2 Hearing
3 Illness
4 Missing Limbs
5 Other

Road Defects

0 No road defects
1 Holes or ruts
2 Shoulder
3 Loose material
4 Construction
5 Other

Possible Violations

0 No apparent violations
1 Had been drinking
2 Reckless driving
3 Speed too fast for conditions
4 Fail to yield right-of-way
5 Improper Passing
6 Improper Backing
7 Improper Turn
8 Fail to Signal
9 Other

TABLE 1-IV-XX

INTENT

Code	Vehicle Intent
01	Go Straight Ahead
02	Overtake
03	Make Right Turn
04	Make Left Turn
05	Make U Turn
06	Slow or Stop
07	Start in Traffic Lane
08	Start from Parked Position
09	Back
10	Remain Stopped in Traffic Lane
11	Remain Parked

Code	Pedestrian Intent
01	Crossing at Intersection or in Crosswalk
02	Crossing not at Intersection or Crosswalk
03	Walking in Roadway with Traffic
04	Walking in Roadway Against Traffic
05	Standing in Roadway
06	Pushing or Working on Vehicle in Roadway
07	Other working in Roadway
08	Playing in Roadway
09	Other in Roadway
10	Not in Roadway
11	Not Stated

TABLE 1-IV-XXI

BODY STYLE

Code	Body Style
01	Passenger Car
02	Mini Bus/Van
03	Bus
04	School Bus
05	Pickup
06	Truck/Truck Tractor
07	Motor Home
08	Motor Cycle
09	Ambulance
10	Farm Tractor/Machinery
11	Construction Machinery
12	Pickup with Camper
13	Bicycle
14	Snowmobile
15	Other

TABLE 1-IV-XXII

TRAILER STYLE

Code	Trailer Style
0	No Trailer
1	Camping Trailer
2	Mobile Home
3	Utility Trailer
4	Boat Trailer
5	Semi Trailer
6	Commercial Cargo Trailer
7	Other

OS JCL Statements

Because HIS is designed to run in conjunction with the IBM System/360 Operating System, the Operating System must receive instructions from the user. Four cataloged procedures containing OS JCL are available for the Accident Subsystem: HISACCM, HISACCF, HISACC, and HISACCA. Each of these contains the necessary JCL for performing certain tasks. The examples in the chapter show which procedure should be used in each instance.

HIS Commands

For HIS commands discussion, including continuation and comment cards, see Chapter 1-II.

Accident file maintenance procedures -- Accidents are keypunched and loaded into the Detail and Vehicle files on demand. Prior to loading, the cards are edited for possible errors. No accidents may be included in the file until they have successfully completed an edit check. After all of the accident data has passed the edit phase, a Detail and a Vehicle file is created. The records in these edit files are in identical format to those in the complete files. Finally, the edit files are merged with the larger three-year Detail and Vehicle files.

EDIT-DATA-CARDS COMMAND:

:EDIT-DATA-CARDS

Editing of accident data cards is performed by program EDIT-DATA-CARDS. The input data set is the group of accident cards being edited, and is defined by DD statement EDITIN. The output data set is defined by DD statement EDITOUT. Any data found to be in error are flagged as they are written into the output data set. This data must be corrected prior to loading into the accident files. Errors detected during the edit run are classified into three categories: terminal errors, severe errors, and warnings. Terminal errors cause immediate rejection

of an entire accident. The entire accident must be re-keypunched. Only one terminal error presently exists. It arises when an accident is punched with an accident number identical to that of another accident, and the program is unable to assign a new number. The error message consists of the entire "A" card, followed by the characters '*** DUPLICATE ACCIDENT NUMBER.' Severe errors are detectable errors which cannot be corrected by the edit program. The data cards for the accident are written into the output file, but are flagged as containing errors. If an attempt is made to load an accident into the accident files prior to correcting severe errors, the accident is rejected. Table 1-IV-XXIII lists the severe error messages and the user's response to the error. Warnings are errors which the edit program may correct after detection. The message is printed to accomplish two purposes: 1) in order that the correction may be verified, and altered if the attempted correction is in error, and 2) in order that the accident report files at the Highway Patrol office may be updated. If the edit program provides the proper correction no user response is required. The accident may be successfully loaded into the accident files as they stand. Table 1-IV-XXIV lists the warnings and the edit program's response. As can be seen from the edit checks, the edit program provides a reasonably complete set of cross-checks on the data cards. Of course, many additional checks remain that could be added to the program to ensure valid data. The edit program is hence likely to be a dynamic program, growing as new checks are found to be desirable.

During the first edit run, the applicable accident data is read, edited, and placed into an ouput file. The user must define two files: the EDITIN file for the accident data cards, and the EDITOUT file for the edit run output. An example of a first edit run for January, 1972 is:

```
//EDIT EXEC HISACCM
//EDITIN DD UNIT=2501,DCB=(BLKSIZE=80,RECFM=F,BUFNO=5)
//EDITOUT DD UNIT=SYSDA,VOL=SER=231499,DISP=(NEW,KEEP),
//           SPACE=(CYL,(6,2)),DSNAME=JAN72.ACCIDENT
//SYSIN DD *
:EDIT-DATA-CARDS
/*
```

TABLE 1-IV-XXIII

SEVERE ERROR MESSAGES AND USER RESPONSES

	Error Message	Probable Cause	User Response
1.	***"A" CARD MISSING	The first card in the set of data cards for an accident does not have an "A" or an "E" in column 1.	Repunch the entire card with the proper code in column 1.
2.	*** SEQUENCE ERROR	A "C" card is not followed by a "D" card, a "D" card is not preceded by a "C" card, or a character other than "A" through "I" appears in column 1 of a data card.	Insert the missing card or correct the invalid character in column 1.
3.	NON-NUMERIC CHARACTER IN COLUMN	A non-numeric character was detected in a field which must contain numeric values.	Repunch columns 1 through 14 of the offending card along with any columns in error.
4.	CITY (25-27) TOO LARGE	The city number field in columns 25-27 of the "A" card contains a value larger than 126.	Repunch columns 1 through 14 of the offending card along with any columns in error.
5.	COUNTY (28-29) TOO LARGE OR ZERO	The county number field in columns 28-29 of the "A" card contains a value larger than 56 or zero.	Repunch columns 1 through 14 of the offending card along with any columns in error.
6.	NUMBER OF VEHICLES (51-52) IS ZERO	The number of vehicles field, columns 51-52 of the "A" card, contains a value of zero. This error will arise only if no "C"-"D" card sequence for vehicles have been coded for the accident.	Repunch columns 1 through 14 and columns 51-52 of the "A" card. A "C"-"D" card sequence must be punched for each vehicle involved in the accident, and these cards inserted into the set of cards for the accident.

TABLE 1-IV-XXIV

WARNING MESSAGES

Warning	Probable Cause	Edit Program's Response
1. YEAR (19-20) IN ERROR	The year in which the accident occurred, coded in columns 19-20 of the "A" card, does not match the first two digits of the accident number.	The first two digits of the accident number are copied in columns 19-20.
2. BOTH VEH (50-51) AND PED (54-55) NUMBER CODED	A "D" card contains a value in both the vehicle number and the pedestrian number field.	Neither value is altered; however, only the vehicle number field is processed.
3. NEITHER VEH (50-51) NOR PED (54-55) NUMBER CODED	A "D" card contains a zero value in both the vehicle number and the pedestrian number field.	Neither value is altered; however, the "C"-"D" card sequence is assumed to indicate a pedestrian.
4. NUMBER OF VEHICLES (51-52) IN ERROR	The number of vehicles coded in columns 51-52 of the "A" card does not match the number of "C"-"D" card sequences supplied defining vehicles.	The value on the "A" card is altered to agree with the number of "C"-"D" card sequence present defining vehicles.
5. NUMBER OF PEDESTRIANS (53-54) IN ERROR	The number of pedestrians coded on the "A" card does not match the number of "C"-"D" card sequences present for pedestrians.	The value on the "A" card is altered to force a match.
6. NUMBER OF FATALITIES (55-56) IN ERROR	The number of fatalities coded on the "A" card does not match the number indicated on the "D" and "I" cards for the accident.	The value on the "A" card is altered in order to force a match.

TABLE 1-IV-XXIV (continued)

Warning	Probable Cause	Edit Program's Response
7. NUMBER OF INJURIES (57-58) IN ERROR	The number of injuries coded on the "A" card does not match the number indicated on the "D" and "I" cards for the accident.	The value on the "A" card is altered in order to force a match.
8. DAMAGE SEVERITY (47) IN ERROR	The damage severity coded on the "A" card does not match the most severe damage code on the vehicle "D" cards.	A match is forced by altering the value on the "A" card.
9. INJURY SEVERITY (46) IN ERROR	The accident injury severity coded on the "A" card does not match the most severe injury severity code on the "D" and "I" cards for the accident.	A match is forced by altering the value on the "A" card.
10. DUPLICATE ACCIDENT NUMBER	An accident contains a number identical to another accident supplied within the same run.	A new number is assigned to the accident.
11. CITY NUMBER (25-27) CODED IN RURAL ACCIDENT	A city number appears in an accident containing a milepoint in the Federal Aid system.	The city number is set to zero.
12. COUNTY (28-29) AND CITY (25-27) DISAGREE	The city specified is not located in the specified county.	The county number is set to the county of the specified city.

Edit runs after the first data edit run consists of two steps: an update step on the accidents, and an edit step. The update step requires three DD statements: one for the input of data cards (CORIN), one for the input of accident data (EDITOUT), and one for the updated output (EDITIN). The edit run requires two DD statements: one for input (EDITIN) and one for output (EDITOUT). The input data set for this run is the output data set of the update step. The output data set is the same data set used for the first edit run; this data set is overwritten. An example of a subsequent run for January, 1972 is:

```
// EXEC HISACCM
//EDITOUT DD UNIT=SYSDA,VOL=SER=231499,DISP=SHR,DSN=JAN72.ACCIDENT
//EDITIN  DD UNIT=SYSDA,SPACE=(CYL,(6,2))
//CORIN   DD *
    update cards
/*
//SYSIN DD *
:UPDATE-ERRORS
:EDIT-DATA-CARDS
/*
```

UPDATE-ERRORS COMMAND:

:UPDATE-ERRORS

During an edit run, accidents containing warnings and severe errors, as well as those containing no detected errors, are saved in an output file. UPDATE-ERRORS allows the capability of updating accidents in which severe errors and improperly-corrected warnings occurred. To allow all of the updating capabilities required for the correction of an edit run, three functions are implemented. First, records may be deleted from the output file. Second, records may be inserted into the output file. Finally, records existing in the file may be revised. In order to perform updates, it is necessary to be able to identify the various cards in the file. This is done by means of the card code, sequence number, and accident number of the data cards -- the first 14 card columns.

This 14-character identifier is herinafter referred to as the "key." To delete a record, the key is coded in columns 1-14 of a data card, and a "delete" character (¬) in column 15. The remainder of the card is left blank. In order to insert a record, it is necessary to identify the location within the file at which the record is to be inserted. This is done by specifying the key of a record that already exists in the file, and a code indicating whether the new record is to be inserted prior to or following the specified record. The data card for inserting a record contains the key of the record already existing in the file in columns 1-14. An asterisk (*) is coded in column 15 if the record is to be inserted before the specified record. A percent sign (%) is coded in column 15 if the record is to be inserted after the specified record. The record being inserted requires an entire data card. This card is placed immediately following the insertion data card. It may occur that two or more records must be inserted at the same location. To accomplish this, code a record with a "greater than" sign in column 1, and blanks in all other columns. This card is placed after the insertion card. The records being inserted are placed after the "greater than" card. Another "greater than" card is placed after the last record. An example of inserting four data cards after record C1720001660201 is:

```
C1720001660201%
>
first data card
second data card
third data card
fourth data card
>
```

The revision function is broken into two sub-functions. The first sub-function allows the revision of all fields of the record other than the key field. The second allows the revision of the key. When using the first sub-function, the key is coded in columns 1-14. Any columns to be altered are coded in the

corresponding card column to the original data card format. For example, if the number of vehicles field of an "A" card is to be altered, the key is coded in columns 1-14 and the number of vehicles in columns 51-52. The remainder of the card is left blank, unless other fields are also being altered. When using the second sub-function, the key as existing on the record is coded in columns 1-14. An equal sign (=) is coded in column 15, and the key to be substituted for the existing key is coded in columns 16-29. Updating of the edit output file is essentially a merge operation. The file is read sequentially along with the data cards supplied for updating and the updates performed as the specified records are encountered. For this reason, it is essential that the data cards be coded and read into the computer in the same order as the records are located in the file. This order may be ascertained from the edit run listing -- place the data cards in the same order as the corresponding records appear in the error listing.

STORE-DATA-CARDS COMMAND:

```
:STORE-DATA-CARDS
```

After the final edit run, the accidents are copied into a tape file for storage. Any accidents flagged as errors are not copied. An example of data card storage for January, 1972 is:

```
// EXEC HIS1
//TAPEOUT DD UNIT=TAPE,VOL=SER=006308,DISP=(NEW,KEEP),
//           DSNAME=JAN72.ACCIDENT,LABEL=(2,RETPD=365)
//EDITOUT DD UNIT=SYSDA,VOL=SER=231499,DISP=SHR,
//           DSNAME=JAN72.ACCIDENT
//SYSIN   DD *
:STORE-DATA-CARDS
/*
```

LOAD-ACCIDENT-DATA COMMAND:

:LOAD-ACCIDENT-DATA

After the final edit run, file EDITOUT contains all of the accident data cards. LOAD-ACCIDENT-DATA converts the data cards into format for loading into the accident files, and places the records in files ACIDENTM and ACCVEHM. In addition to performing the conversions, a file is built containing the data for the Highway Patrol memos. This file must be defined by DD statement MEMOS. After loading, the accidents are merged into the full accident files by program MERGE-ACCIDENT-FILES and stored on tape by STORE-DATA-CARDS. These two steps must be completed before attempting to load the following edit run, as the records in ACIDENTM and ACCVEHM will be overwritten during the next load operation. An example of data card loading for January, 1972 is:

```
// EXEC HISACCF
//EDITOUT DD UNIT=SYSDA,VOL=SER=231499,DISP=SHR,
//           DSNAME=JAN72.ACCIDENT
//MEMOS   DD UNIT=SYSDA,VOL=SER=231498,DISP=(NEW,KEEP),
//           SPACE=(CYL,(2,1)),DSNAME=JAN72.MEMOS
//SYSIN   DD *
:LOAD-ACCIDENT-DATA
/*
```

MERGE-ACCIDENT-FILES COMMAND:

:MERGE-ACCIDENT-FILES

After loading, the vehicle and detail files formed must be sorted and merged with the indexed-sequential files. MERGE-ACCIDENT-FILES performs the merge operation. The IBM sort-merge routine is used for the sort. An example of the merge operation for January, 1972 is:

```
// EXEC HISSORTA
//SORTIN DD DISP=OLD,DSNAME=HIS.ACIDENTM
//SORTOUT DD DISP=OLD,DSNAME=HIS.ACIDENTM
//SYSIN DD *
  SORT  FIELDS=(2,12,A),FORMAT=CH,SIZE=E1000
  END
/*
// EXEC HISSORTA
//SORTIN DD DISP=OLD,DSNAME=HIS.ACCVEHM
//SORTOUT DD DISP=OLD,DSNAME=HIS.ACCVEHM
//SYSIN DD *
  SORT  FIELDS=(2,15,A),FORMAT=CH,SIZE=E2000
  END
/*
// EXEC HISACCF
//SYSIN DD *
:MERGE-ACCIDENT-FILES
/*
```

Summary commands -- The Accident summary generating programs provide the capability of producing the Montana Highway Patrol's Accident Memos and the National Safety Council's Form 16.

PRINT-MEMOS COMMAND:

```
:PRINT-MEMOS
```

After loading a group of data cards, the memos may be printed from the MEMOS file built by LOAD-ACCIDENT-DATA. The PRNTMEMO DD statement must be included in order to place the memos on the desired paper form. An example of memo printing for January, 1972 is:

```
// EXEC HIS1
//PRNTMEMO DD SYSOUT=(0,,2462)
//MEMOS   DD UNIT=SYSDA,VOL=SER=231498,DISP=SHR,
//           DSN=JAN72.MEMOS
//SYSIN   DD *
:PRINT-MEMOS
/*
```

NATIONAL SAFETY COUNCIL FORM 16 COMMAND:

```
:FORM-16,START-DATE=nn/nn/nn,
:         END-DATE=nn/nn/nn,LOCATION= { ALL       }
                                      { city-name }
```

After completing the file maintenance procedures, the National Safety Council Form 16 report may be run. The beginning date and ending date is specified on the command; any length of time from 1 day up may be specified. The LOCATION parameter allows the production of the Form 16 report for statewide accidents, and for cities. When LOCATION=ALL is specified, all reportable accidents in the state during the specified time period are included. When LOCATION=city-name is specified, all accidents occurring in the city, regardless of amount of damage, are included. When coding the name of a city in the LOCATION parameter, blanks must be replaced with hyphens. For example, Great Falls is coded as GREAT-FALLS. An example of Form 16 commands and OS JCL for January, 1972 is:

```
// EXEC HISACC
//SYSIN DD *
:FORM-16,START-DATE=01/01/72,END-DATE=01/31/72,
:         LOCATION=ALL
:FORM-16,START-DATE=01/01/72,END-DATE=01/31/72,
:         LOCATION=GREAT-FALLS
/*
```

Report commands -- In order to produce the annual accident-by-sections report, the Accident Detail file, Roadlog file, Traffic file, and True Mileage file must all be utilized. Cataloged procedure HISACCA contains DD statements for all of these files, and may be used when producing the report. Accidents are stored in the Detail file according to accident number. In order to access the file by location, as required for the report, a Directory file is generated. In order to save a second access to the Detail file, all of the information required for the report is also placed in the Directory file. The Traffic file is used for defining the highway sections for the report. A skeleton version of an Accident Report file is generated from the

Traffic file and Roadlog file, defining the sections and storing the Roadlog descriptions, number of lanes, and city numbers. A second pass through the Accident Report file utilizes the Directory file already created, and fills in accident data (such as the number of accidents occurring in each section). During the same pass, the number of lanes is copied from the Report file into the Directory file. A final pass through the Accident Report file utilizes the Traffic and True Mileage files, and fills in the traffic counts for the sections. In the accident-by-sections listing, high and low accident rates are flagged with asterisks (*). To determine the rates to be flagged, average rates must be calculated for each route, and an upper and lower limit established. An Accident Limits file is built containing these limits.

CREATE-DIRECTORY and LOAD-ACC-DIRECTORY COMMANDS:

:CREATE-DIRECTORY

:LOAD-ACC-DIRECTORY

The Accident Directory file is built from the Accident Detail file, and contains one record for each accident occurring on an Interstate, Primary or Secondary route having a milepoint coded. Hence, no records are placed in the directory for the following situations:

1. Accidents occurring on a Primary or Secondary route in a city, as the accident is located by city coordinates rather than by reference post.

2. Accidents occurring on Secondary routes which are not reference posted, are located by range, township, and section rather than by reference post.

3. Accidents occurring off the Federal Aid system.

CREATE-DIRECTORY builds a sequential file containing the location, accident number, date, time, number of injuries, number of fatalities, first harmful event, collision type, and road surface condition fields of the Detail file. The resultant file is sorted on the location and accident number fields,

providing a unique key field. LOAD-ACC-DIRECTORY then loads the sequential file into an indexed-sequential file. An example of the Job Control statements and HIS commands for creating the Directory file is:

```
//CREATE EXEC HISACCA
//SYSIN DD *
:CREATE-DIRECTORY
/*
//SORT EXEC HISSORTA
//SORTIN DD UNIT=DISK,VOL=SER=231415,DISP=OLD,DSN=HIS.ACCDIR
//SORTOUT DD UNIT=DISK,VOL=SER=231415,DISP=OLD,DSN=HIS.ACCDIR
//SYSIN DD *
  SORT  FIELDS=(2,25,A),FORMAT=CH,SIZE=E15000
  END
/*
//LOAD EXEC HISACCA
//SYSIN DD *
:LOAD-ACC-DIRECTORY
/*
```

CREATE-ACCSUB COMMAND:

```
                          ⎧ SECTIONS ⎫
:CREATE-ACCSUB,PHASE=     ⎨ ACCIDENT ⎬
                          ⎩ TRAFFIC  ⎭
```

The Report file is generated in three passes, or phases. The first phase utilizes the Traffic and Roadlog files to define the sections and fill in the Roadlog data. The second phase utilizes the Accident Directory file to fill in accident data (this phase also completes the Directory file with the Roadlog number of lanes). The third phase utilizes the Traffic and True Mileage files to fill in traffic counts. An example of the Job Control statements and HIS commands for creating the Accident Report file is:

```
//CREATE EXEC HISACCA
//SYSIN DD *
:CREATE-ACCSUB,PHASE=SECTIONS
:CREATE-ACCSUB,PHASE=ACCIDENT
:CREATE-ACCSUB,PHASE=TRAFFIC
/*
```

Note: The Accident Directory file must be created and loaded before attempting to create the Accident Report file.

<u>CREATE-ACC-LIMITS COMMAND</u>:

```
:CREATE-ACC-LIMITS
```

The Accident Limits file is built from the Report file by program CREATE-ACC-LIMITS. After all three phases of CREATE-ACCSUB have been successfully executed, the Limits file may be built. An example of the Job Control statements and HIS command necessary to build the file is:

```
//CREATE EXEC HISACCA
//SYSIN DD *
:CREATE-ACC-LIMITS
/*
```

<u>ACCIDENT-BY-SECTIONS and MULTIPLE-ACC-LOCNS COMMANDS</u>:

$$\text{:ACCIDENT-BY-SECTIONS,REPORT=ACCIDENT,DATA=}\left\{\begin{array}{l}\text{INT}\\ \text{PRIM}\\ \text{SEC}\end{array}\right\}$$

```
:MULTIPLE-ACC-LOCNS,REPORT=ACCIDENT
```

After all of the CREATE-ACCSUB phases have been completed, the Accident-by-Sections report may be printed. The first portion of the report is a listing, by section, of the accident rates. The second portion of the report is a summary by multiple accident location, in which all roadways are divided into tenths-of-a-mile. For locations having two or more accidents, all accidents are listed. Program SYS-PARAM is invoked to indicate to HIS that pages are to be numbered for the report. FORMAT=NOREDUCE specifies the number of lines printed per page, as well as the positioning for the page numbers. An example of the Job Control statements and HIS commands for printing the report is:

```
// EXEC HISACCA
//SYSIN DD *
:SYS-PARAM,FORMAT=NOREDUCE,PAGE-NUMBER=1
:ACCIDENT-BY-SECTIONS,REPORT=ACCIDENT,DATA=INT
:ACCIDENT-BY-SECTIONS,REPORT=ACCIDENT,DATA=PRIM
:ACCIDENT-BY-SECTIONS,REPORT=ACCIDENT,DATA=SEC
:MULTIPLE-ACC-LOCNS,REPORT=ACCIDENT
/*
```

Storage of the Accident Data

All of the Accident data is stored on magnetic disk packs for access by the HIS System. As the data is transferred from the user's coding formats of Tables 1-IV-I and 1-IV-XVII to magnetic disk, the formatting of many of the data fields are altered to enhance storage and execution efficiency. For details regarding the internal record formats, refer to Highway Information System Volume 2: Programmer Information.

CHAPTER 1-V
SUFFICIENCY USER INFORMATION

Introduction

The Montana Department of Highway's Planning and Research Bureau has adopted a sufficiency rating system in order to compare the sufficiency of the existing rural highways with the latest design standards. This rating system takes into consideration the structural adequacy, the safety, and the traffic capacity of the highway.

The sufficiency file is a disk-resident file which contains the sufficiency rating information for Montana's Federal Aid Interstate and Primary Highways. However, the file has been designed to store sufficiency rating information for the Federal Aid Secondary systems should this data become available.

Sufficiency records contain information about a section of highway rather than about a point on the highway. A "Sufficiency section" is identified by specifying the beginning milepoint of the section. Section breaks begin at any point along the highway where a significant change in the highway's structural adequacy, safety, or traffic capacity occurs.

There are two major catagories of records stored in the Sufficiency file: rating records and descriptor records. Sufficiency rating and descriptor records are constructed on disk from data cards submitted by the user. All file updating is accomplished through file maintenance which is explained in later sections of this chapter.

Sufficiency Rating Record Coding

Sufficiency rating records contain the structural and safety ratings for sections of a highway. The coding format for the sufficiency rating records is shown in Table 1-V-I.

Sufficiency Descriptor Record Coding

There are four types of sufficiency descriptor records: 1) Municipal, 2) Coincident, 3) Out-of-State, and 4) End of Route. Only two fields are

TABLE 1-V-I

DATA ELEMENTS AND CODING INFORMATION
FOR SUFFICIENCY MILEAGE RECORDS

Column(s)	Item	Remarks
1	F. A. Route System	See Table 1-II-II.
2-4	F. A. Route Number	See Note 1 of Table 1-II-I.
5-7	Reference Post	See Note 1 of Table 1-II-I.
8	Plus Sign	Code "+" in this field.
9-13	Distance	See Note 2 of Table 1-II-I.
14-31	Description	Verbal Description of section.
32	Rural/Urban	Coded, but not used in present subsystem.
33-34	Design Speed	See Note 1 below.
35	Terrain	See Note 2 below.
36-37	Average Speed	See Note 3 below.
38-39	Percent of Sight Distances less than Design	See Note 4 below.
40-41	Number of Stopping Sight Distances less than Design	See Note 5 below.
42-43	Number of Horizontal Curves Sharper than Design Degree of Curvature	See Note 6 below.
44	Number of Narrow Bridges	See Note 7 below.
45-46	Foundation Rating	See Note 8 below.
47-48	Surface Rating	See Note 9 below.
49-50	Drainage Rating	See Note 10 below.

Notes: 1. The Design Speed for the Interstate System and designated portions of other Primary highways is 70 mph. For the Primary and Secondary Highways classified as arterials, the normal range of design speed as influenced by terrain conditions is from 40 to 60 mph. These terrain conditions are:

Terrain	Design Speed
Level	60
Rolling	50-60
Mountainous	40-50

Frequent variation in design speeds on a sequence of rating sections is undesirable. Judgment must determine whether terrain conditions require reduced speeds by reason of non-feasibility of construction to higher standards. The following are design controls in relation to design speeds:

Design Speed (mph)	Max. Curve (degree)	Sight Distance (ft.) Stopping	Sight Distance (ft.) Passing
50	7-1/2	350	1700
60	5	475	2000
70	3	600	2300

In addition, if the section of roadway being rated is under construction or non-existent, two special design speeds are coded. "01" is the design speed coded for a section of highway under construction. Zero (00) is the design speed if the section is non-existent.

2. Terrain is stored as one of the following digits:

Plains	1
Rolling	2
Mountainous	3

The classification "Plains" means that no appeciable portion of the rating section contains natural features that would require extensive grading to attain minimum design requirements. "Rolling" country is that in which desirable design could feasibly be obtained even though a certain amount of heavy construction would be required. A "Mountainous" classification indicates terrain of such character that design must be restricted considerably because of the non-feasibility of the construction effort required to raise the design speed.

3. The average highway speed is the weighted average of the design speeds within a highway section, when each subsection within the section is considered to have an individual design speed.
4. Sight distance is the distance visible to the driver of a passenger vehicle, measured along the normal travel path of a roadway, to the roadway surface or to a specified height above the roadway, when the view is unobstructed by traffic. For sections of two-lane primary routes and secondary arterials it is necessary to ascertain the percent of the total rating-section length in which the road surface is not visible to the driver of a passenger car for a distance of at least 1500', because of either horizontal or vertical curvature.
5. Stopping sight distance is the distance required by a driver of a vehicle, traveling at the design speed, to bring his vehicle to a stop after an object on the roadway becomes visible. It includes the distance traveled during the perception and reaction times and the vehicle braking distance.
6. The design degree of curvature is the greatest central angle subtended by a 100' length of curve which can be safely negotiated by a vehicle traveling at the highway design speed. The number of horizontal curves sharper than the design degree of curvature indicates the number of curves within the sufficiency section which have a degree of curvature greater than the design degree of curvature.
7. A narrow bridge is any bridge within the sufficiency section which is narrower than the traveled-way width.
8. A foundation can only be rated 10 for being adequate or 0 for being inadequate. A rating of 0 is given to primary sections if any one of the following conditions are present in the section:

    (a) traveled-way less than 18' wide;
    (b) lack of adequate and uniform cross section, including side ditches; or

(c) paved surface indicating failure which could not be corrected by the addition of a few inches of surfacing material.

For secondary sections, a rating of 0 is given if any one of the following conditions are present in the section:

(a) traveled-way is less than 18' wide;
(b) not graded to reasonably adequate and uniform cross section, including side ditches;
(c) unpaved road showing signs of becoming impassable in adverse weather; or
(d) paved surface indicating failure that could not be corrected by the addition of a few inches of surfacing material.

For local-service routes a rating of 0 is given if any one of the following conditions are present in the section:

(a) traveled-way less than 14' wide;
(b) unpaved road showing signs of becoming impassable in adverse weather; or
(c) paved surface indicating failure that could not be corrected by the addition of a few inches of surfacing material.

9. The highway surface is given a rating from 0 to 30. Certain controls pertaining to the remaining life of the pavement have been established to insure a consistent surface rating system. When a paved surface is in relatively good condition, but showing first signs of failure (cracks, raveling, etc.), it should be rated 15. A surface of more advanced failure, while the road is still in fair and usable condition, will be rated between 10 and 15. A rating of 10 indicates that the pavement is in a condition that makes replacement incipiently justifiable. Increasingly poor conditions to the point of complete deterioration, creating a hazard, are rated 10 to 0. Rating above 15 is graduated in relation to surface smoothness, surface conformance to proper crown and grade, and uniformity of these conditions throughout the rating section. It should be kept in mind that the rating applies to the entire rating section. One short pavement failure (which can be remedied by a minor maintenance operation) should not significantly affect the section rating, therefore the rating is proportioned on the basis of the section length and failed length. Any considerable extent of variation in condition justifies an additional rating section. Gravel roads are rated only from 0 to 10 according to thickness, quality, and condition of the surfacing material. Unimproved, bladed, and graded and drained surface types are rated 0.
10. The sufficiency sections are rated from 0 to 10 depending on the adequacy of drainage facilities. Lack or inadequacy of drainage facilities reduces the total of 10 points allotted for complete drainage. The amount of reduction depends on the proportion of the total section length affected and the seriousness of the condition.

coded on a data card for each descriptor record. First the milepoint field, consisting of the F. A. System, F. A. Route, and the reference post (see Table 1-II-II and 1-II-I), is coded in columns 1-13. The second field is coded in columns 14-31. This second field contains a formatted description of the sufficiency section. If the section of highway runs through a municipality, the user codes the word "CITY" in columns 14-17 and columns 18-31 are left blank. If the section of highway is coincident with a previously rated route, the user codes the word "COINCIDENT" in columns 14-23 and columns 24-31 are left blank. For the sections of a highway which are not within the state's boundaries the beginning milepoint and the description "OUT-OF-STATE" followed by 6 blanks is coded. The End of Route record describes the ending milepoint of a Federal Aid Route. The user codes the ending milepoint in columns 1-13 and the description "END OF ROUTE" followed by 6 blanks. It is important for the Sufficiency user to note that the milepoint coded for each of the descriptor records must correspond with the coincident, municipal, out-of-state, and end of route records stored in the Roadlog file (see Table 1-II-XI).

OS JCL Statements

In order to execute any of the programs in the Sufficiency Subsystem the user must submit user commands and input data using the following Job Control Language:

```
// EXEC HISSUFF,TIME=n
//SYSIN DD *
     user commands and input data
/*
```

"// EXEC HISSUFF,TIME=n" informs the IBM Operating System that the user will be executing programs within the Sufficiency Subsystem and that the Subsystem has been allotted "n" minutes to execute these programs. "//SYSIN DD *" indicates that user commands and input data are being submitted. "/*" signals the end of user commands and input data.

HIS Commands

For HIS Commands discussion, including continuation and comment cards, see Chapter 1-II.

Report and Summary Commands -- The first step in the reporting of the sufficiency ratings is to gather and store the structural, safety, and traffic capacity data for the Interstate and Primary Highways. Five separate commands must be executed to retrieve this data and store it in the Sufficiency Report file, SUFFSUB. These commands are:

```
:CREATE-SUFFSUB,PHASE=SUFFICIENCY
:CREATE-SUFFSUB,PHASE=ROADLOG
:CREATE-SUFFSUB,PHASE=TRAFFIC
:CREATE-SUFFSUB,PHASE=ACCIDENT
:CREATE-SUFFSUB,PHASE=CALCULATION
```

After the five CREATE commands have been executed and the Sufficiency Report file has been generated, the user can then submit any of the following Sufficiency Report or Summary commands:

CREATE-SUFFSUB,PHASE=SUFFICIENCY COMMAND:

```
:CREATE-SUFFSUB,PHASE=SUFFICIENCY
```

This user command retrieves the necessary sufficiency data from the Sufficiency file and stores this data in the Sufficiency Report file. The elements retrieved from the Sufficiency file are:

milepoint,
description,
design speed,
terrain,
average highway speed,
percent of sight distance less than 1500 feet,
number of stopping sight distances less than design,
number of horizontal curves sharper than the design degree of curvature,

number of narrow bridges,
foundation rating,
surface rating,
drainage rating, and
section length.

For a more detailed description of these data elements see Table 1-V-I. An example of the CREATE-SUFFSUB,PHASE=SUFFICIENCY command and the accompanying OS JCL statements is:

```
// EXEC HISSUFF,TIME=5
//SYSIN DD *
:CREATE-SUFFSUB,PHASE=SUFFICIENCY
/*
```

CREATE-SUFFSUB,PHASE=ROADLOG COMMAND:

```
:CREATE-SUFFSUB,PHASE=ROADLOG
```

This user command retrieves the necessary roadlog data from the Roadlog file and stores this data in the Sufficiency Report file. The elements retrieved from the Roadlog file are:

year built,
year improved,
surface width,
roadway width,
surface type,
number of lanes,
divided or undivided highway,
city number, and
county number.

For a more detailed description of these data elements see Table 1-II-I. An example of the CREATE-SUFFSUB,PHASE=ROADLOG command and the accompanying OS JCL statements is:

```
// EXEC HISSUFF,TIME=10
//SYSIN DD *
:CREATE-SUFFSUB,PHASE=ROADLOG
/*
```

CREATE-SUFFSUB,PHASE=TRAFFIC COMMAND:

:CREATE-SUFFSUB,PHASE=TRAFFIC

This user command retrieves the necessary traffic data from the Traffic file and stores this data in the Sufficiency Report file. The elements retrieved from the Traffic file are:

average daily traffic,
design hourly volume,
percent commerical traffic, and
current average daily traffic.

The average daily traffic (ADT) used in the sufficiency rating is a weighted average ADT based on the last three consecutive data years. An example follows to illustrate the precise nature of the weighted ADT computation:

As illustrated in Figure 1-V-1, assume that a section to be rated (for sufficiency) has the beginning milepoint "A" and the ending milepoint "C." Also assume that the Traffic file has traffic counts at milepoints "A," "B," and "C." X_1 and X_2 are the "true mileage" distances between milepoints "A" and "C." Assuming the Traffic file contains the last three consecutive years of traffic data for count stations "A," "B," and "C," this traffic data can be designated by a doubly subscripted variable:

Count Station	Corresponding Traffic Data
A	T_{11}, T_{12}, T_{13}
B	T_{21}, T_{22}, T_{23}
C	T_{31}, T_{32}, T_{33}

The mathematical computation of the average weighted ADT for the sufficiency section is:

$$\text{Year 1 weighted ADT} = \frac{X_1(T_{11}+T_{21})/2 + X_2(T_{21}+T_{31})/2}{(X_1+X_2)}$$

$$\text{Year 2 weighted ADT} = \frac{X_1(T_{12}+T_{22})/2 + X_2(T_{22}+T_{32})/2}{(X_1+X_2)}$$

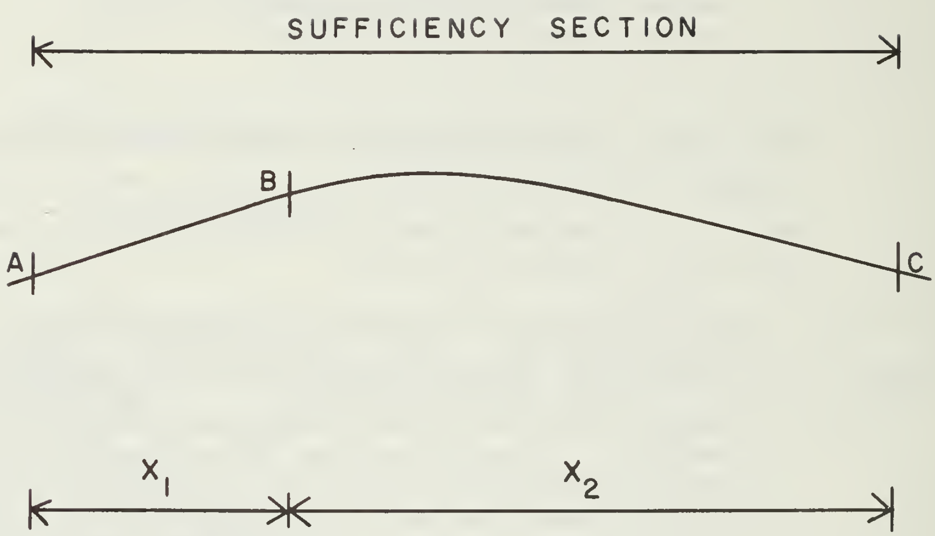


Figure 1-V-1. Traffic counts within a sufficiency section.

$$\text{Year 3 weighted ADT} = \frac{X_1(T_{13}+T_{23})/2 + X_2(T_{23}+T_{33})/2}{(X_1+X_2)}$$

Average Weighted ADT =
(Year 1 weighted ADT + Year 2 weighted ADT + Year 3 weighted ADT)/3

If the traffic counts do not occur at the same reference points as the sufficiency breaks, an interpolated weighted average daily traffic for the sufficiency breaks is calculated. Also, if the traffic data for the sufficiency section does not contain counts for the last three consecutive years the Average Weighted ADT is modified by averaging only the yearly weighted ADT's available.

The maximum percent design hourly volume residing in the records of the Traffic file for the sufficiency section is multiplied by the Average Weighted ADT (as computed above) and the result is placed in the Sufficiency Report file as the design hourly volume. After placing the design hourly volume in the Sufficiency Report file the Sufficiency Subsystem retrieves the percent commercial traffic from the same traffic data record. The current average daily traffic is the Average Weighted ADT based on the last data year. An example of the CREATE-SUFFSUB,PHASE=TRAFFIC command and the accompanying OS JCL statements is:

```
// EXEC HISSUFF,TIME=10
//SYSIN DD *
:CREATE-SUFFSUB,PHASE=TRAFFIC
/*
```

<u>CREATE-SUFFSUB,PHASE=ACCIDENT COMMAND</u>:

```
:CREATE-SUFFSUB,PHASE=ACCIDENT
```

This user command retrieves the average number of accidents occurring within the sufficiency section for the last three years. The Sufficiency Subsystem accesses the Accident Directory file which contains a record of those accidents which have occurred on the Federal Aid System within the last three years. The Subsystem then computes the average number of accidents

occurring within each sufficiency section. An example of the CREATE-SUFFSUB,PHASE=ACCIDENT command and the accompanying OS JCL statements is:

```
// EXEC HISSUFF,TIME=5
//SYSIN DD *
:CREATE-SUFFSUB,PHASE=ACCIDENT
/*
```

<u>CREATE-SUFFSUB,PHASE=CALCULATION COMMAND:</u>

```
:CREATE-SUFFSUB,PHASE=CALCULATION
```

The user command "CREATE-SUFFSUB,PHASE=CALCULATION" executes the procedures used to calculate the sufficiency ratings and store these calculations along with the data in the Sufficiency Report file. The sufficiency ratings calculated for the Report file are:

safety rating,
capacity rating,
total rating,
adjusted rating, and
deficient mileage.

The safety rating is a function of the hazardous roadway conditions and the number of accidents occurring within the sufficiency section. The safety rating formula is:

$$\text{Safety Rating} = \frac{(2 \times \text{section length})}{(N_1 + N_2 + N_3 + N_4)}$$

where: section length = the sufficiency section length (from the Sufficiency Report file);

N_1 = the number of stopping sight distances less than that permitted by the design (from the Sufficiency Report file);

N_2 = the number of horizontal curves sharper than the design speed will permit (from the Sufficiency Report file);

N_3 = the number of narrow bridges per sufficiency section (from the Sufficiency Report file); and

$$N_4 = \frac{\text{number of accidents per sufficiency section} \times 10^7}{\text{average weighted ADT per sufficiency section} \times 365 \times 100}$$

(The number of accidents in the section is the average number of accidents based on the last three years and is retrieved from the Sufficiency Report file. The Average Weighted ADT is from the Sufficiency Report file.) (The safety has maximum value of 20.)

The capacity rating formula is:

$$\text{Capacity Rating} = 30 - (15 \times \text{Design Hourly Volume})/\text{Service Volume}$$

(Both the design hourly volume and the service volume for the sufficiency section are stored in the Sufficiency Report file.)

The total rating is the summation of the foundation, surface, drainage, safety, and capacity ratings. The maximum value for the total rating (an adequate section) is 100. The percent of deficient mileage is the total rating subtracted from 100. The deficient mileage is the percent of deficient mileage for the sufficiency section multiplied times the sufficiency section length. An adjusted sufficiency rating is computed to give additional priority to sections of highway carrying the larger traffic volumes. The equation for the adjustment is:

$$\text{Adjusted Rating} = \text{Total Rating} + \frac{(\text{Total Rating})^2 - (100 \times \text{Total Rating})}{50 \times \text{Log(avg. system ADT)}} \times (\log(\text{avg. section ADT}) - \log(\text{avg. system ADT}))$$

(The average current ADT for the entire highway system is extracted from the Traffic Summary file. The current average daily traffic for the sufficiency section is taken from the Sufficiency Report file.)

LIST-BY-SECTION COMMAND:

$$\text{:LIST-BY-SECTION,REPORT=SUFFICIENCY,DATA=} \left\{ \begin{array}{l} \text{INT} \\ \text{PRIM} \\ \text{SEC} \\ \text{INT=n} \\ \text{PRIM=n} \\ \text{SEC=n} \\ \text{INT=n-n} \\ \text{PRIM=n-n} \\ \text{SEC=n-n} \end{array} \right\}$$

"LIST-BY-SECTION" provides a list of the Rural Federal Aid Routes and their sufficiency section ratings. Two examples of the user command and the accompanying OS JCL statements are:

```
// EXEC HISSUFF,TIME=10
//SYSIN DD *
:LIST-BY-SECTION,REPORT=SUFFICIENCY,DATA=PRIM=3-8
:LIST-BY-SECTION,REPORT=SUFFICIENCY,DATA=INT
/*
```

LIST-BY-DISTRICT COMMAND:

```
:LIST-BY-DISTRICT,REPORT=SUFFICIENCY
```

"LIST-BY-DISTRICT" is a listing of the sufficiency sections in each of the twelve financial districts in the state. The summary begins with Financial District Number 1 and continues in ascending order through Financial District Number 12. This summary uses a sorted copy of the Sufficiency Report file. The Sufficiency Subsystem is used to copy and print the Sufficiency Report file for this summary. The sorting is done by an IBM Utility Program. The following example lists the Job Control statements and the user commands for copying, sorting, and printing the summary:

```
// EXEC HISSUFF
//SYSIN DD *
:COPY-FOR-SORTING,REPORT=SUFFICIENCY
/*

// EXEC HISSORTA
//SORTIN DD DISP=OLD,DSN=HIS.SUFFREP
//SORTOUT DD DISP=OLD,DSN=HIS.SUFFREP
//SYSIN DD *
     SORT FIELDS=(36,2,A,85,3,A),FORMAT=CH,SIZE=E1200
     END
/*

// EXEC HISSUFF
//SYSIN DD *
:LIST-BY-DISTRICT,REPORT=SUFFICIENCY
/*
```

LIST-BY-RATING COMMAND:

```
:LIST-BY-RATING,REPORT=SUFFICIENCY
```

"LIST-BY-RATING" is a list of the sufficiency sections in order of the percent of deficient mileage in each sufficiency section. The summary begins with those sections having the largest percentage of deficient mileage. This summary uses a sorted copy of the Sufficiency Report file. The Sufficient Subsystem is used to copy and print the Sufficiency Report file for this summary. The sorting is done by an IBM Utility Program. The following example lists the Job Control statements and the user commands for copying, sorting, and printing the summary:

```
// EXEC HISSUFF
//SYSIN DD *
:COPY-FOR-SORTING,REPORT=SUFFICIENCY
/*

// EXEC HISSORTA
//SORTIN DD DISP=OLD,DSN=HIS.SUFFREP
//SORTOUT DD DISP=OLD,DSN=HIS.SUFFREP
//SYSIN DD *
     SORT FIELDS=(85,3,A),FORMAT=CH,SIZE=E1200
     END
/*

// EXEC HISSUFF
//SYSIN DD *
:LIST-BY-RATING,REPORT=SUFFICIENCY
/*
```

MAP-TABLES COMMAND:

```
                                              ⎧ INT       ⎫
                                              ⎪ PRIM      ⎪
                                              ⎪ SEC       ⎪
                                              ⎪           ⎪
                                              ⎪ INT=n     ⎪
:MAP-TABLES,REPORT=SUFFICIENCY,DATA=          ⎨ PRIM=n    ⎬
                                              ⎪ SEC=n     ⎪
                                              ⎪           ⎪
                                              ⎪ INT=n-n   ⎪
                                              ⎪ PRIM=n-n  ⎪
                                              ⎩ SEC=n-n   ⎭
```

"MAP-TABLES" provides a listing of the sufficiency sections by Rural Federal Aid Routes in a table format. These tables are used to describe mapped sections of the Rural Federal Aid Routes within the Sufficiency By Sections Report. An example of MAP-TABLE commands and the accompanying OS JCL statements is:

```
// EXEC HISSUFF,TIME=10
//SYSIN DD *
:MAP-TABLES,REPORT=SUFFICIENCY,DATA=PRIM
:MAP-TABLES,REPORT=SUFFICIENCY,DATA=INT=15
/*
```

RATING-BY-DISTRICT COMMAND:

```
                                                    ⎧ INT       ⎫
                                                    ⎪ PRIM      ⎪
                                                    ⎪ SEC       ⎪
                                                    ⎪           ⎪
                                                    ⎪ INT=n     ⎪
:RATING-BY-DISTRICT,REPORT=SUFFICIENCY,DATA=        ⎨ PRIM=n    ⎬
                                                    ⎪ SEC=n     ⎪
                                                    ⎪           ⎪
                                                    ⎪ INT=n-n   ⎪
                                                    ⎪ PRIM=n-n  ⎪
                                                    ⎩ SEC=n-n   ⎭
```

"RATING-BY-DISTRICT" prints a table of financial districts and the percent of sufficient mileage within each financial district. Examples of RATING-BY-DISTRICT commands and the accompanying OS JCL statements are:

```
// EXEC HISSUFF
//SYSIN DD *
:RATING-BY-DISTRICT,REPORT=SUFFICIENCY,DATA=INT+PRIM
:RATING-BY-DISTRICT,REPORT-SUFFICIENCY,DATA=PRIM=30-50
/*
```

DEF-MILES-BY-COUNTY COMMAND:

```
                                                ⎧ INT       ⎫
                                                ⎪ PRIM      ⎪
                                                ⎪ SEC       ⎪
                                                ⎪           ⎪
                                                ⎪ INT=n     ⎪
:DEF-MILES-BY-COUNTY,REPORT=SUFFICIENCY,DATA=   ⎨ PRIM=n    ⎬
                                                ⎪ SEC=n     ⎪
                                                ⎪           ⎪
                                                ⎪ INT=n-n   ⎪
                                                ⎪ PRIM=n-n  ⎪
                                                ⎩ SEC=n-n   ⎭
```

"DEF-MILES-BY-COUNTY" summarizes the amount of deficient mileage within each county. Examples of the DEF-MILES-BY-COUNTY command and accompanying OS JCL statements are:

```
// EXEC HISSUFF
//SYSIN DD *
:DEF-MILES-BY-COUNTY,REPORT=SUFFICIENCY,DATA=PRIM
:DEF-MILES-BY-COUNTY,REPORT=SUFFICIENCY,DATA=INT
/*
```

Sufficiency file maintenance commands -- The structural adequacy, safety, and traffic capacity of rural highways are continually changing. In order to maintain an up-to-date Sufficiency file the Sufficiency Subsystem provides the user with the ability to update, list, copy and backup the Sufficiency file.

UPDATE COMMAND:

```
                                   ⎧ DELETE  ⎫
:UPDATE,FILE=SUFFICIENCY,FUNCTION= ⎨ INSERT  ⎬ , DDNAME=ddname
                                   ⎩ REWRITE ⎭
```

Delete, insert, and rewrite are the three methods of updating the Sufficiency file. "ddname" is a user supplied name that the

Sufficiency Subsystem can refer to for data describing the records in the file which are to be updated. The DELETE option informs the Sufficiency Subsystem that the user wishes to delete complete records from the Sufficiency file. One data card must be coded for each record being deleted. The data card contains the record key in columns 1-13 and the remainder of the data card is left blank. The record having the corresponding key will be deleted from the file. If no record with the specified key exists in the file, an error message is printed. The INSERT option indicates that the user wishes to add new records to the Sufficiency file. A complete data card with all of the appropriate fields coded must be supplied for each record to be inserted. The format of this data card is shown in Table 1-V-I. Before this new record is inserted the coded fields are edited to insure that the new data added to the file is within the limits described in Table 1-V-I. If any errors are detected an error message is printed and the record is not inserted. The REWRITE option is used to update specific fields within an existing record in the Sufficiency file. The fields are changed by coding the record key and the field or fields to be rewritten. Coding dollar signs ($) in any data field will cause the corresponding fields to be filled with blanks. Before the new fields are rewritten they are edited. If any errors are detected in the fields to be rewritten, an error message is printed and the field or fields are not rewritten. From time to time the user may find it necessary to rewrite the record key. If so, the user must submit a separate rewrite data card with the existing key coded in columns 1-13, an equal sign (=) in column 14, and the new key in columns 15-27. The Subsystem then deletes the existing record and inserts a new record with the new key. The following example illustrates the OS JCL statements and user commands necessary for updating the Sufficiency file:

```
// EXEC HISSUFF,TIME=3
//SYSIN DD *
:UPDATE,FILE=SUFFICIENCY,FUNCTION=DELETE,DDNAME=OUT
:UPDATE,FILE=SUFFICIENCY,FUNCTION=INSERT,DDNAME=IN
/*

//OUT DD *
    data cards describing the records to be deleted.
/*

//IN DD *
    data cards describing the new records to be inserted.
/*
```

COPY COMMAND:

$$:\text{COPY,FILE=SUFFICIENCY,LIST=} \left\{ \begin{matrix} \text{YES} \\ \text{NO} \end{matrix} \right\}$$

This command copies the entire Sufficiency file into a backup area. The list option allows the user to list the contents of the file as it is being copied. Because the backup file may be placed on a tape or disk chosen by the user, the backup area must be defined to the IBM Operating System. This is done by means of a DD statement named SAVESUF. A complete discussion of OS Job Control Language for defining the backup area is beyond the scope of this manual; refer to the IBM publication: OS Job Control Language Reference Manual. An example of the COPY command and the accompanying OS JCL statements is:

```
// EXEC HISSUFF,TIME=5
//SAVESUF DD UNIT=TAPE,VOL=SER=012345,DISP=(NEW,KEEP),
             DSNAME=HIS.SUFFICIENCY.BACKUP,
             DCB=(BLKSIZE=640,LRECL=64,RECFM=FB)
//SYSIN DD *
:COPY,FILE=SUFFICIENCY,LIST=YES
/*
```

CREATE COMMAND:

$$:\text{CREATE,FILE=SUFFICIENCY,LIST=} \left\{ \begin{array}{c} \text{YES} \\ \text{NO} \end{array} \right\}$$

This command may be used to restore a permanent Sufficiency file from a backup file. The list option allows the user to list the file as it is created. In addition to the command, the user must supply the IBM Operating System with a DD statement which indicates where the backup file is located. An example of the CREATE command and the accompanying OS JCL statements is:

```
// EXEC HISSUFF,TIME=5
//SAVESUF DD UNIT=TAPE,VOL=SER=012345,DISP=OLD,
             DSNAME=HIS.SUFFICIENCY.BACKUP
//SYSIN DD *
:CREATE,FILE=SUFFICIENCY,LIST=YES
/*
```

CHAPTER 1-VI

PRELIMINARY STUDY OF RETRIEVAL OF ROADWAY GEOMETRIC INFORMATION

Introduction

The purpose of this study was to investigate the feasibility of constructing a file describing the horizontal and vertical geometry of roadways in Montana.

Information describing the geometric characteristics of a test section of roadway, located south of Livingston, was obtained from as-built plans and was used to build a computer disk file of the centerline geometry. To check the integrity of the data, a program was written to compute the relative coordinates between locations on the test section. The test section was composed of four roadways, forming a loop, so that closure of the traverse could be tested. The closure of this arbitrarily selected traverse can provide an indication whether such a data base is feasible for many envisioned uses.

In this study, the roadway was considered to consist solely of its centerline, with the geometric data defining the path of the centerline in three-dimensional cartesian coordinates. The problem was divided into two parts: 1) the horizontal (x-y plane), and 2) the vertical (z coordinate). The horizontal projection of the centerline consists solely of straight lines (tangents), segments of circles (arcs), and portions of spirals. The vertical profile consists of tangents and portions of parabolas.

Roadway Locations

Two methods are presently in use in Montana for the specification of roadway locations along a route: 1) the reference post system, and 2) the route stationing system.

The reference post system -- The reference post system on Montana Federal Aid routes always locates points on a route with respect to the beginning of that route. The route begins with reference post zero and succeeding reference posts are placed in numerical sequence approximately a mile apart

along the route. The HIS True Mileage file indicates the exact distance from the beginning of the route to each reference post. To specify a roadway location (milepoint), a reference post and a distance from the reference post to the milepoint are given. The distance from the reference post may be more or less than one mile because of the non-uniform spacing and is measured from the reference post in the direction of increasing reference post numbering.

The stationing system -- The stationing system consists of a set of reference points (stations) uniformly spaced 100 feet apart, and is used primarily in the layout and construction of the roadway. Right-of-way monuments are normally tied to this system but they are rarely marked with stationing information. Because the stationing information is not physically marked in the field, it is necessary to have the as-built plans of the route available if these monuments are to be used for reference points. The necessity of having the as-built plans available when using this system greatly limits field use of the system by anyone except engineers and surveyors. An additional problem in keeping a sequential set of stations occurs when a portion of a route is realigned, causing a change in roadway length. If the rebuilt route were to keep a sequential set of stations, then changes in stationing would have to be made for all points beyond the roadway realignment. These changes in stationing would require large amounts of extra work and hence are not done. Instead a "station-to-station equation" is utilized to change from one reference point to another. For example, a route may consist of 100 stations 100 feet apart. The roadway is rebuilt between stations 20 and 30, and in the process, the road is lengthened by 200 ft. Assigning two new stations along the route would result in having 102 station points; earlier references to stations 20-100 would no longer be valid. In practice, however, the original points 1 through 20 are left intact, as are stations 30-100. New stations 21-32 are assigned to the new section of roadway. Duplicate stations 30, 31, and 32 now exist; a station-to-station equation is supplied equating the first station 32 with the second station 30; these stations are at identical locations.

Relationship between the reference post and stationing systems -- The files in the HIS (Highway Information System) data bank are based on the reference post system. All roadway locations are specified with respect to the physical reference posts. This system is advantageous because all locations on the route may be specified by referring to a unique reference post (as opposed to the stationing system, in which a single route may have several stations with the same number). The stationing system, allowing distance discrimination to 1/100th of a foot, is used in roadway planning and construction and other engineering applications where it is necessary to pin-point locations to that precision. Because of the necessity of accuracy in using geometric data the "base" geometric data is taken from as-built plans which are based on the stationing system. The ideal situation for the geometrics data base will consist of the use of both systems, with the computer performing conversions from one to the other. Utilizing the reference post system for storage of data (the "key") will allow access to the Roadlog and other files. Utilizing the stationing system for computations allows the accuracy required.

Geometrics Files

For this study two files were constructed: 1) a horizontal data file, and 2) a vertical data file. Each record in both files contains a "location" field consisting of the route system, the route number, and the stationing. The stationing is written as the station number followed by a positive distance from the station to the point in question. For example, stationing 10434.8 defines a point 34.8 feet from station 104 toward station 105.

The length of a section is determined by comparing the stationing at the beginning of the section (given in the record defining that section) to the stationing at the beginning of the next section (given in the record defining the next section). For example, if one section begins at station 10325.3 and the next begins at 12248.5, the section length is 100x(122-103) + (48.5-25.3), or 1923.2 feet.

The horizontal file -- The horizontal file consists of five types of records, each with different types of information. The record types are

identified by a record type code stored within the record. The five types are:

Type code	Record Type
E	Equation
I	Intersection
T	Horizontal Tangent
C	Horizontal Curve
F	Final Route Entry

Equation records contain station-to-station equations (allows the programs to correct for discontinuities in station numbering). The contents of this record type are:

1. location at renumbering point,
2. type code E, and
3. ahead stationing.

Intersection records are not used in geometric computations, but identify the locations of intersections with other routes. The contents of this record type are:

1. location,
2. type code I, and
3. location on intersecting route (route system, route number, and stationing).

Horizontal tangent records describe roadway sections which are horizontal tangents. In order to perform computations on horizontal tangents, the length of the section must be known. The length is calculated by comparing the stationing at the beginning of the section to the stationing at the beginning of the next section. If an equation record occurs between two sections, it must be taken into account in calculating the length. (Intersection records are ignored in calculating the length.) If a tangent ends the route, a final route entry record is included to give the stationing at the end of the tangent. The bearing coded in a horizontal tangent record is the direction of the tangent. For example, a tangent heading 36 degrees 39 minutes and 30 seconds

east of due north is coded as N 36 39 30 E. The contents of a horizontal tangent record are:

1. location of beginning point of section,
2. type code T, and
3. bearing.

Horizontal curve records describe roadway sections which are horizontal curves. Because stationing continues around curves (segments of circles), the length of the arc is calculated as described above for the horizontal tangent. When proceeding through a route, the bearing is calculated and tallied from record to record. Hence, the bearing at the beginning of the arc will be known when the record is read. (Note: when a route begins with a curve, the bearing can be specified by means of a zero-length tangent record appearing ahead of the curve record.) The central angle is specified in terms of degrees, minutes, and seconds. The direction of curve is an alphabetic code, 'R' if the curve bends to the right, and 'L' if to the left. The contents of these records are:

1. location of beginning point of section,
2. type code C,
3. central angle of the arc, and
4. direction of curve.

The final route entry record signals the end of the data included for that route. Its presence is necessary in order to calculate the length of the last section of the route (whether it be a tangent or curve). The contents of the record are:

1. location, and
2. type code F.

<u>The vertical file</u> -- The vertical file consists of five types of records, each with different types of information. The record types are identified by a record type code stored within the record. The five types are:

Type Code	Record Type
E	Equation
I	Intersection
T	Vertical Tangent
C	Vertical Curve
D	Equation in Grade

The equation and intersection records are identical to those described in the horizontal file. Vertical tangent records describe roadway sections whose profiles are straight lines (tangents). The projected horizontal section lengths are calculated in the same fashion as are the horizontal tangents, using the stationing of the first vertical tangent or vertical curve record following the tangent record. The grade is specified as a positive percentage for ascending grades, and a negative percentage for descending grades. The contents of these records are:

1. location of beginning point of section,
2. type code T, and
3. grade.

Vertical curve records describe roadway sections whose profiles are vertical curves (parabolas). In performing computations with vertical curves, it is necessary to know the slope at each end of the curve. This value is found on vertical tangent records; therefore a vertical tangent record must precede and follow each vertical curve record. When a vertical curve immediately follows another vertical curve on a roadway, a zero-length vertical tangent record must be placed in the file between the two vertical curve records to give the grade at that point. The projected length of the vertical curve is found by comparing stationing on the vertical curve record with that on the vertical tangent record following the vertical curve record, exactly as is done on the case of horizontal tangents. The contents of these records are:

1. location, and
2. type code C.

Equation-in-grade records are used for corrections between roadway plans and the actual roadway built. Equations-in-grade are adjustments for differences that occur when different benchmarks are used for individual projects, and when the projects are connected the equation-in-grade establishes the equivalency. An equation-in-grade is recorded on the as-built plans to reflect the difference. The difference in elevation is given in feet, and is positive if the difference must be added to the calculated elevation, and negative if the difference must be subtracted. The contents of the records are:

1. location
2. type code D, and
3. difference in elevation.

Geometrics Programs

The initial programming was accomplished using the Montana Department of Highway's IBM System 360 Model 40 computer. Once the compatability of the data with the reference posting system had been checked and some test files built and edited, it was decided to test the data on a series of roadways forming a loop using the Montana State University's Xerox Data Systems Sigma 7 computer tc facilitate ease of accessability.

The initial program on the IBM computer was written in PL/I programming language; however, the XDS computer does not have a PL/I compiler. Therefore, the rest of the programs were written in the FORTRAN programming language. Only the FORTRAN programs used on the XDS computer are described in this report.

Vertical program -- The initial objective of the vertical program was to calculate the relative difference in elevation between any two points on the roadway centerline. The program had to be modified somewhat from this initial objective because the stationing did not describe a unique point on each route due to the presence of station equations. The program presently will only calculate the relative difference in elevation between a point defined by the first such occurrence of a station on a route and another

point that is marked by a vertical data record entry. A method of avoiding this limitation is to describe each point of the road uniquely by means of its milepoint as is done with other programs now in use by the Montana Department of Highways.

Horizontal program -- The program to calculate differences in horizontal locations calculates distances with respect to compass directions. A positive Y distance is a distance North while a negative Y distance is a distance South. Similarily, a positive X distance is a distance East and a negative X distance is a distance West. Like the vertical program, the horizontal program calculates relative X and Y distances between specified start and end locations. The start location is the first such designated stationing encountered in the sequential file for that route. The program assumes that the end location is the first record encountered, in a sequential search, with a designation common to the end location specified.

Special conditions for the programs -- Both the vertical and horizontal computer programs have inherent assumptions. One is that both the starting and the ending locations must be on the same route. The second assumption is that when a start location is specified, and it occurs at the same location as a stationing equation, it is assumed that the start location is the back station of the station-to-station equation. The third assumption is that whenever the end location occurs at the exact location of a stationing equation, the end location is the ahead station of the station-to-station equation.

Geometrics Test Section

The test section is located between Livingston and Gardner. A sketch of the test section is shown in Figure 1-VI-1.

The test section consists of portions of Federal Aid Primary route 11, Federal Aid Secondary route 476, Federal Aid Secondary route 540, and Federal Aid Secondary route 362. Pertinent information about these sections of roadway are:

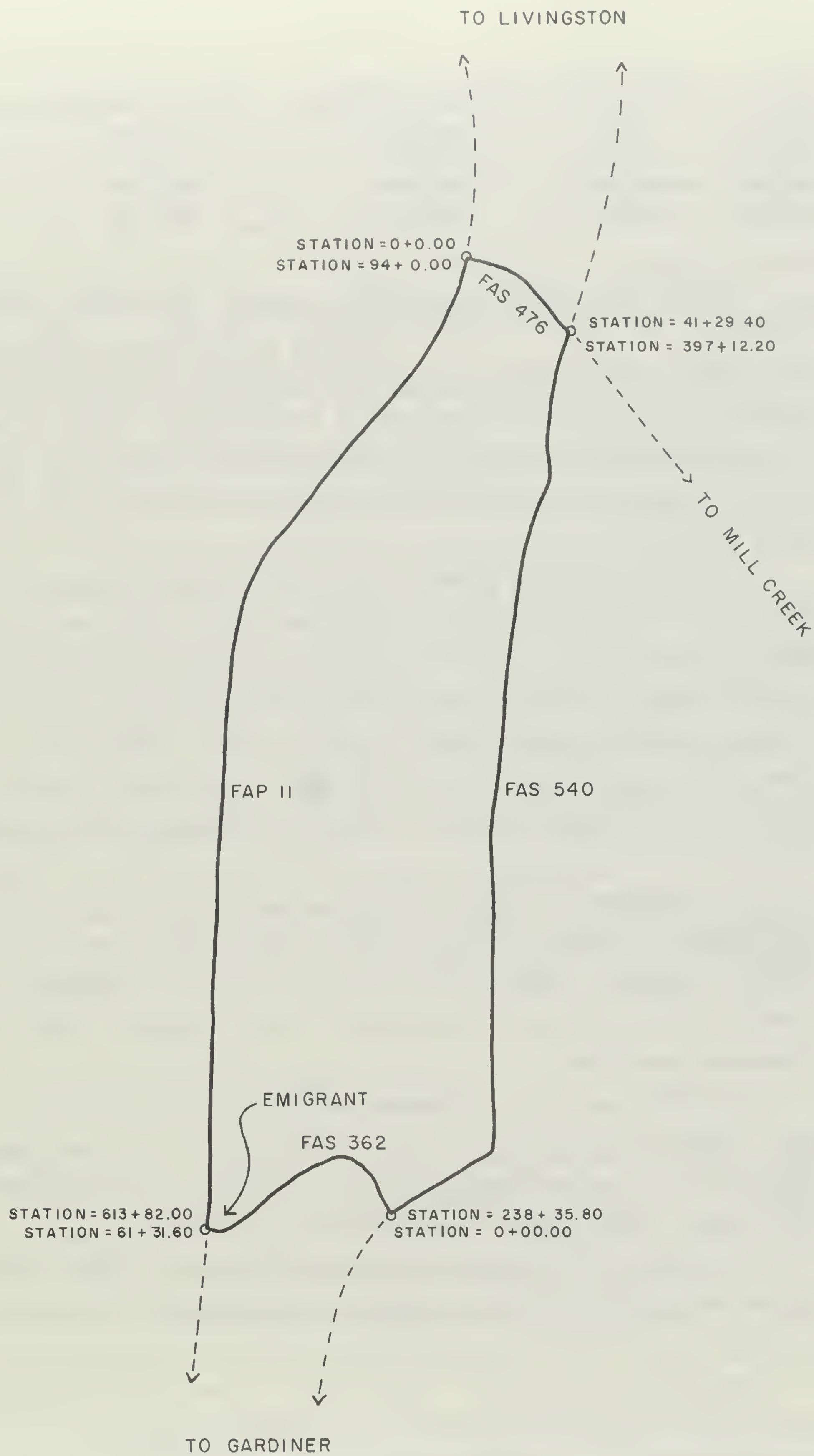


Figure 1-VI-1. Sketch of test section.

Route	Year Built	Starting Milepoint	Ending Milepoint	Section Length
FAP11	1961	31+0.010	37+0.190	6.178 miles
FAS476	1961	0+0.000	0+0.777	0.777 miles
FAS540	1926,29,41	14+0.720	20+0.400	5.700 miles
FAS362	1961	0+0.000	1+0.160	1.164 miles

By picking a corner of the loop, say the south-west corner, and specifying, in order, the start and end locations on: FAP11, FAS476, FAS540, and FAS362, the vertical program will calculate the relative elevations from one end to the other and keep a running total. The start and end locations of these routes are defined as the intersection locations. They are not the true beginning and ending locations of the routes, but for these considerations they are.

The total distance around the loop was calculated to be 72,826.81 feet. The total elevation difference should have been zero, but was -1.15 feet. These results didn't take into account equations-in-grade appearing on the as-built plans, because these records were not placed in the file. The total of the equations-in-grade comes to +1.63 feet. Adding this to the total elevation, we get an error of +0.48 feet. In other words, in a traveled distance of approximately 13.75 miles the vertical distance checks within 1/2 foot.

The horizontal calculations indicated a total of 72,826.81 feet for the distance traveled, which exactly matched the distance calculated by the vertical program. Running both the vertical and horizontal programs provides a check on the data, but in addition it also provides a good check on the stationing equations.

The total relative X distance should have been zero, but was -137.86 feet. The total relative Y distance should also have been zero, but was +142.25 feet. The straight-line error distance to the starting point is 198.09 feet. This gives a relative error of 1:370.

As a check on the horizontal FORTRAN programs, COGO programs were run on the test sections and the results compared within one foot.

Problems encountered -- One of the problems encountered in coding the data from the as-built plans was in finding the actual intersection point of two roadways. A common intersection was given at the shoulder line of

the "through" roadway. The plans provided insufficient information to ascertain if the common station given for the "through" roadway was a right-angle station at the shoulder-centerline intersection or the actual station where the centerlines of both roadways intersected. By using plane geometry and the angle of intersection, a "through" station was calculated at the centerline-centerline intersection, assuming the common station was in fact a shoulder line-centerline inter-section station.

Another problem encountered was the necessity of determining the actual elevation of the intersection of two roadways. The beginning and ends of two connecting roads were at the 12 foot shoulders. It was necessary to assume 12 foot of normal crown (.02 foot/foot) to arrive at a common intersection elevation. In other cases, where two roadways intersected at a curve of one, the superelevation from the centerline to the shoulder was used.

An irregular profile was encountered on one roadway. A grade raise, for cover over a culvert, was necessary. This was done during construction and it deviated from normal in that the grade line was raised from a point behind the vertical curve to intersect the plan grade in the middle of the vertical curve. The vertical data was coded as the plan grade and therefore is inaccurate throughout the length of the grade raise. This is probably a very unique situation that is not normally encountered.

An error was found on the "as-built" plans. An angle was actually to the right, but was labeled left. This error showed up in the bearing of the following tangent.

Recommendations Concerning Geometrics File

Although the data gathering process to get the information necessary to construct a geometrics file for the roadways in Montana will be expensive, it appears inevitable that such a file will be necessary at some future date.

Present information systems restrict considerations to the roadway only and each route is considered as a separate and lineal system. A properly constructed geometrics file and related software would offer the following advantages:

1) The separate routes could be geometrically related so that travel over the entire system could be simulated. This could make "routing" problems easier and faster.

2) Stationing and reference posting could be related to the State Plane Coordinate System by establishing "ties" at appropriate locations. It seems that this would be most helpful in relating the roadway system to items of interest off the roadway, such as material sources, land ownership, land use, etc.

3) The inevitable change to the metric system could be made easier.

4) Geometry of roadway segments would be readily available for comparison with other data bases. For example, an analysis of geometrics and its relationship to accidents could be made.

5) Safety considerations, such as sight-distances, could readily be investigated.

It should be kept in mind that a geometrics file should consist of more than just the geometry of the centerline. It is imperative that cross-sectional information also be available in a geometrics file, as well as objects off the roadway that offer a driver visual obstruction.

CHAPTER 1-VII

STORAGE AND RETRIEVAL OF VISUAL INFORMATION

Introduction

Preliminary consideration has been given to the development of a retrieval system for visual information to be used in conjunction with the HIS data files. The retrieval system would contain supplementary data which is not easily or economically stored in digital computer files.

The immediate application prompting the consideration of "visual files" is the storage and retrieval of the sketches on the accident reports that are needed in accident analyses. Many man-days that could more profitably be spent in accident analysis, if source document images were readily retrievable, are now being expended in searching for source documents.

The desirable system should have several attributes, including:

1) Direct access -- It should not be necessary for a user to have to scan visually through several feet of microfilm or physically sort through dozens of aperture cards in order to access the desired image.

2) Hardware compatibility -- It would be desirable if the user could access the visual file in the same manner that he will be accessing the digital file -- with a user-oriented command system. It should also be pointed out that the digital and visual files should be complementary and not competitive, therefore hardware compatibility would not prohibit interfacing the digital and visual files.

3) Updating the visual file should be a one-step operation. This criterion focuses on the desireability of a central system for storing images, rather than having a central processing center physically distributing new and updated images to remote storage centers.

4) Remote access -- A user should be able to access a central storage file without having to walk or drive blocks or miles to do so. The user's time can best be spent in deciding what data he needs and in analyzing that data -- not in acquiring it.

5) Simultaneous access -- More than one user should be able to access the file at the same time. Whether this simultaneous access should

be extended to include simultaneously accessing the same image within the file is an economic consideration.

Several "methods" of storage and retrieval of visual information have been given preliminary consideration with three basic concepts involved. These three basic concepts are:

1) digitally-stored images,
2) microfilm methods, and
3) computer graphics methods.

Digitally-stored images are very nice, but the storage requirements are excessive when considering the volume of images that will eventually be stored. In the system reviewed, it is required that a 4,096 x 4,096 matrix be stored to represent numerically a single image. Essentially each element of the matrix represents the light intensity of a dot, with 4,096 x 4,096 dots making up the image. The system was more than likely developed for temporary image storage and enhancement rather than for permanent image storage and retrieval. Because of the apparent high costs of capital equipment for processing and storage it does not appear that this system is a feasible solution.

Microfilm and computer graphics methods appear to have merit, not only in satisfying immediate needs -- but also in serving long-term needs, if planned properly. Because there are several microfilm and computer graphics alternatives, each concept will be presented along with a discussion of the advantages and disadvantages of each.

Microfilm Methods

There are three basic microfilm retrieval methodologies presently marketed, each having its peculiar advantages and disadvantages. Each method is described and the advantages and disadvantages are discussed in the following paragraphs.

It is presumed in each case that the production of the microfilm image has already been accomplished, inasmuch as that process must be accomplished for each of the methods. It is further presumed that any selected retrieval device would have the capability of producing a "hard copy" of the microfilmed image on user command.

The real differences in the methods, when considering information retrieval are those mentioned previously: 1) direct or non-direct access of images, 2) hardware/software compatibility with digital files, 3) methods of updating information, 4) physical relationship between the user and the files (remote or near access), and 5) simultaneous access to files and/or images by the users.

Roll/sheet/aperture card microfilm -- For the purposes of this report, the "roll," "sheet," and "aperture card" microfilm methods will be considered in the same category inasmuch as they are basically the traditional methods of microfilm storage and retrieval. The source documents are photographed onto microfilm and the microfilm roll or sheet contains many frames of source documents whereas the aperture card generally contains only one frame. These rolls, sheets, or aperture cards are then visually scanned, through the use of a magnifying reader in order to access the sought-after frame of information. The general advantages of this broad concept are:

1) it is relatively inexpensive, and

2) the equipment is simple to operate.

The general disadvantages are:

1) it does not allow for direct access of images and a roll, sheet, or card must be physically selected and placed in the reading device;

2) there is no hardware or software for interfacing and therefore the systems are not in a true complementary state;

3) physical distribution of microfilm rolls, sheets, or aperture cards must be made to each access location;

4) updating requires "file clerks" or "film splicers" at every access location and requires making referral notes on the outdated images; and

5) there would be a strong tendency to withhold distribution of information until rolls or sheets were filled.

It is recognized that the aperture card method, because it generally contains only one source image, is not as difficult to update as are the rolls and sheets, but the physical replacement of the document at each access location complicates file maintenance.

"Keyed microfilm" -- For the purposes of this report a "keyed microfilm" system is one in which information may be coded between the image frames. The coded information usually contains a reference number identifying the image following the coded block along with other digital information pertaining to that image. The coded information is optically scanned so that rapid sequential scanning of a roll of microfilm to access the desired frame may be accomplished. In addition, the coded block may be accessed, or information in the coded blocks may be accumulated and/or compared. Other than this unique code block for each image, which demands a rather sophisticated set of hardware, the system is the same as the "roll microfilm." The advantages and disadvantages are identical to those of the roll microfilm except that access is faster, once the roll is mounted. The other information in the coded block would be redundant if digital files are available for the information being stored.

Central microfilm with remote access -- The principal difference between this system and other microfilm systems is that the microfilm image is resident in an automated central file and all access is to this one file through hard-wired terminals with the image appearing to the user on a TV screen. File volume does not seem to be a problem because a single "unit" can store up to 200,000 aperture cards. Depending on micrographic compression of the source documents, this single "unit" could store up to 11,000,000 pages of information. The time delay between entering the request on the keyboard at the remote station and the appearance of the image on the TV screen is reported to be a maximum of six seconds. Depending on the configuration of the hardware, the system has the capability of being interfaced with the digital computer through which the digital file is accessed. Scan and zoom features on the remote TV make possible magnification of any part of the image. The system disadvantages are:

1) it is relatively expensive,

2) all users are dependent on one central file (it is conceivable that this could create hardships if the system were "down" for an extended period of time), and

3) the remote stations must be hard-wired to the central file.

The system advantages are:

1) access is direct to the image desired;

2) hardware is compatible with present generation digital computers allowing users to "simultaneously" access the digital and visual files;

3) there is only one file to keep updated and each image, generally residing in an aperture card, can easily be updated;

4) remote access is superior; and

5) simultaneous access by more than one user is possible.

It is suggested that site visits to operating microfilm installations by intended system users would be very important before further study or evaluation is made. The corporations marketing the systems have expressed interest in cooperating with personnel of the Montana Department of Highways by scheduling visits to agencies using their respective equipment on a day-to-day basis.

Computer Graphics Methods

The storage of relatively simple visual information through graphic terminals should be given serious consideration. Sketches of accidents are relatively simple graphic displays, and the storage of this information through graphic terminals could tremendously enhance accident analysis capability.

The principal reason for this possible analysis enhancement is because each element of the sketch would be stored digitally along with its relative location at the accident scene. Examples of elements that would be stored by relative location are: roadway or intersection, traffic control devices, vehicles or pedestrians involved, vegetation, and other contributing physical objects.

Admittedly data input would be more complicated than microfilming, but the end-product could well be worth the increased effort. The hardware costs have dropped by almost an order of magnitude in recent months so that hardware cost, in itself, does not appear to be prohibitive.

With regard to accident analyses, for example, it would appear that once such a file is created that software could be developed to "draw" collision diagrams.

Computer graphics methods of storing selected visual information should be considered on a continuing basis.

<u>Summary</u>

In all probability there is need, on a long-term basis, for multiple methods of storing "visual" information. The methods that evolve will depend to a large degree on the intended use of the "visual file."

If the need is purely security (to have backup records in the event of a fire), then perhaps simple roll microfilm with simple magnifying readers will suffice. If the desire is to access the information for operations or administration (payroll records produced on computer-outputed-microfilm, or right-of-way plots) uses then the central, remotely accessed, file should be considered. Finally, if the visual file has information conducive to meaningful analyses, then a computer graphics file may well be worthy of consideration.